Mutah University
College of Graduate Studies

Computer Network Traffic Classification Using Machine Learning Technique

By
Nosaiba Hamdan Abu-Samhadanh

Supervisor:
Dr. Mouhammd Al-Kasassbeh

A thesis submitted to the Collage of Graduate Studies in partial fulfillment of the requirements for the Master's Degree in Computer Science to the Department of Information Technology, University of Mutah.

Mutah University, 2015

الآراء الواردة في الرسالة الجامعية لا تُعبر بالضرورة عن وجهة نظر جامعة مؤتة

بسم الله الرحمن الرحيم

MUTAH UNIVERSITY
College of Graduate Studies

جامعة مؤتة
كلية الدراسات العليا

نموذج رقم (١٤)

قرار إجازة رسالة جامعية

تقرر إجازة الرسالة المقدمة من الطالبة نسيبه حمدان ابو سمهدانه الموسومة بـ:

Computer network traffic classification using machine learing technique

استكمالاً لمتطلبات الحصول على درجة الماجستير في **الحاسوب.**

القسم: تكنولوجيا المعلومات.

	التوقيع	التاريخ	
د. محمد شراري الكساسبة		٢٠١٥/١٢/٢١	مشرفاً ورئيسا
د. محمد علي العبادي		٢٠١٥/١٢/٢١	عضواً
د. بسام المحادين		٢٠١٥/١٢/٢١	عضواً
د. جعفر صالح القطاونة		٢٠١٥/١٢/٢١	عضواً

عميد الدراسات العليا

د. محمد المحاسنة

MUTAH-KARAK-JORDAN
Postal Code: 61710
TEL :03/2372380-99
Ext. 5328-5330
FAX:03/ 2375694
e-mail: dgs@mutah.edu.jo sedgs@mutah.edu.jo
http://www.mutah.edu.jo/gradest/derasat.htm

مؤتة – الكرك – الاردن
الرمز البريدي: ٦١٧١٠
تلفون: ٩٩-٠٣/٢٣٧٢٣٨٠
فرعي 5328-5330
فاكس 375694 ٠٣/٢
البريد الالكتروني
الصفحة الالكترونية

ACKNOWLEDGEMENTS

In the name of Allah the Merciful

And prayers and peace be upon the best prophets and messengers...

Thanks to Allah for his good graces … Thanks for his blessing and for honoring me with the completion of this work ... and I dedicate it to Allah as beneficial knowledge …

I dedicate this work:

To our prophet Mouhammd … the one whose message has lighted the paths of the world.

To those pure spirits … to the martyrs of the stones and the martyrs of Arab Muslim countries.

To the one who gave me life ... doaa and love ... my dear father.

To the one who has held me with her heart ... and hugged me with her eyes ... and surrounded me with love and doaa … my beloved mother.

My dear parents ... with whom I have lived in their smiling eyes …

To those who have surrounded me with support and concern … have smiled upon my success… my beloved brothers and sisters.

To the one who has stood beside me with his supervision, support and advice … step by step … to my dear Supervisor: Dr. Mouhammd Al-Kasassbeh… and all respected Faculty members.

And lastly, to my friends and to all who have helped me to accomplish this work.

I thank you with all my heart…

Nosaiba Hamdan Abu-Samhadanh

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AA	Average Accuracy Rate.
AdaBoost	Adaptive Boosting.
ARFF	Attribute Relation File Format.
CSV	Comma Separated Values.
DPI	Deep Packet Inspection.
DR	Detection Rate.
FN	False Negative.
FP	False Positive.
FPR	False Positive Rate.
Gtalk	Google talk.
GP	Genetic Programming.
GUI	Graphical User Interface.
HTTP	Hypertext Transfer Protocol.
HTTPs	Hypertext Transfer Protocol secure.
ML	Machine Learning.
MLP	MultiLayer Perceptron.
NIC	Network Interface Card.
Non-VOIP	Non-Voice Over Internet Protocol.
PCE	Process Classification Engine.
PSD	Packet Size Distribution.
P2P	Peer to Peer.
RMSE	Root Mean Squared Error.
ROC	Receiver Operating Characteristic.
SSL	Secure Sockets Layer.
TCP	Transport Control Protocol.
TLS	Transport Layer Security.
TL	Transport Layer.
TN	True Negative.
TP	True Positive.
UDP	User Datagram Protocol.
VOIP	Voice Over Internet Protocol.
WEKA	Waikato Environment for Knowledge Analysis.
WWW	World Wide Web.
Yahoo MSN	Yahoo Messenger.

ABSTRACT

Computer Network Traffic Classification Using Machine Learning Technique

Nosaiba Hamdan Abu-Samhadanh
Mutah University, 2015

In recent years, the uses of the Internet has increased and been extensively developed. Many modern applications have evolved to facilitate the process of social communication. Also the traffic classification process has appeared as a science in itself on the Internet nowadays.

In this thesis, we generate a new dataset and tested it through four Machine Learning (ML) algorithms: Adaptive Boosting (meta.Adaboost (j48)), Random Forest, J48 and MultiLayer Perceptron (MLP). Additionally, we separated the classification process into two cases: Non-Voice Over Internet Protocol (Non-VOIP) and Voice Over Internet Protocol (VOIP) applications, the second one is called Multiclasses, which contains five applications (classes), namely: PayPal, YouTube, Google talk (Gtalk), Yahoo Messenger and Skype.

We choose these applications from the Transport Layer (TL). The generated dataset was compiled by means of a different process that included: packet capturing, features extraction and classification processes, we using also four statistical features. The dataset used here contains real data from a live network using an experimental testbed from experimental environment within a campus environment. In the both cases: Non-VOIP and VOIP case and Multiclasses classification case, the meta.Adaboost (j48) classifier achieved the highest accuracy level among other classifiers, of 98.6605% and 98.3007% respectively. The J48 classifier achieved the minimum time for building the training model in the two cases of classification. Also, the MLP took the maximum time between other classifiers for build the training model in both cases.

الملخص

تصنيف حركة المرور في شبكات الحاسوب باستخدام تقنية تعلم الآلة

نسيبه حمدان أبو سمهدانة

جامعة مؤتة، 2015

في السنوات الأخيرة إزدادت وتطورت إستخدامات الإنترنت. وقد تطورت العديد من التطبيقات الحديثة لتسهيل عملية التواصل الاجتماعي. بحيث أصبحت عملية تصنيف حركة المرور تبدو كعلم في حد ذاته على شبكة الإنترنت في الوقت الحاضر.

في هذه الأطروحة قمنا بإنشاء قاعدة بيانات جديدة وقيمناها من خلال أربعه من خوارزميات تعلم الآله: (meta.Adaboost (j48), Random forest, J48 and MLP). بالإضافة إلى ذلك، قمنا بتقسيم عمليه التصنيف لحالتين: تطبيقات غير صوتيه عبر الانترنت وتطبيقات صوتيه عبر الانترنت. وتدعى الحاله الثانيه ب متعددت الفئات (Multiclasses) والتي تحتوي خمسة تطبيقات (classes) وهي : (PayPal, Gtalk, Yahoo Messenger, Skype and YouTube).

أختيرت هذه التطبيقات من طبقة تسمى ((Transport Layer (TL)). تم إنشاء قاعدة البيانات المنشأه والتي تم جمعها من خلال عمليات مختلفه وهي: إلتقاط البيانات، إستخراج الخصائص وتصنيف البيانات المستخرجه، إستخدمنا أيضا أربع خصائص إحصائيه. تحتوي قاعدة البيانات التي أنشأناها هنا على بيانات حقيقية من شبكة حية تم جمعها باستخدام اختبارات تجريبية من بيئه تجريبيه ضمن بيئة الحرم الجامعي. لحالتي التصنيف (Non-VOIP and VOIP) و (Multiclasses) حقق المصنف ((meta.Adaboost(j48) أعلى مستوى دقة بين المصنفات الأخرى، ونتيجته تساوي 98.6605% و 98.3007٪ على التوالي. بينما حقق المصنف J48 أقل وقت بين المصنفات الأخرى لبناء نموذج التدريب لكل من حالتي التصنيف. وأيضا المصنف MLP إستغرق أكبر وقت من بين المصنفات الأخرى لبناء نموذج التدريب لكلا حالتي التصنيف.

CHAPTER ONE
INTRODUCTION

1.1 Background to the Internet and Traffic Classification

The history of the Internet begins with the development of electronic computers in the 1950s, and ever since then the Internet has become established as a basic, vital need for different types of users. Now it has become very widely used for many different purposes and is developing rapidly, being generally available more and more everywhere in the world. Either via smart phones for individual users, or via computers for companies and government institutions, universities and many other organizations.

The uses of the Internet are many and varied, depending on the user’s requirements. Multiple applications are growing continually, to provide many services for users. The most well-known applications used nowadays are to facilitate communication between people, as well as the different uses for the management of companies and multinational organizations (Alshammari & Zincir-Heywood, 2015). Some of the services available are: videos, calls, chat messengers, file transfer and on-line services applications. Therefore, it is necessary to carry out a classification process, in order to protect the traffic which passes through the networks that are used to connect the different applications. The main goals for the wide use of these applications are to facilitate the business and reduce the time, effort and cost to all Internet users.

People use different applications according to their needs, such as communicating with people over long distances. An application like PayPal can be used to manage financial transactions from home without the need to go personally to a bank. YouTube can provide domestic work opportunities for Internet users by facilitating the establishment of their own channels, sharing videos, etc.

Internet traffic refers to the flow of data across the Internet network. Researchers have recently used the term ‘traffic classification’ to describe the techniques for classifying traffic. Classification is based on the number of features that support the accuracy of the performance depending on the goals of the classification process.

Focus has concentrated on traffic classification in the last few years. The importance of traffic classification has increased with the increase in the development of information and applications and is widely used in many domains such as network management, design, security, research, advertising and communication (Xue, Wang, & Zhang, 2013). For example when describing traffic for specific applications, like Gtalk traffic, the traffic classification is sometimes called traffic identification. Despite the constraints on global Internet traffic because of security concerns, there is

still enormous development potential. This has also prompted the development of traffic classification techniques in parallel (Dainotti, Pescape, & Claffy, 2012).

The efficiency and accuracy of the application classification process represents the keystone of network monitoring, because it is most important for network management (Qin, Wang, Liu, & Guan, 2015). Traffic classification is a very important automatic process that divides network traffic into a number of classes (Xue, Wang, & Zhang, 2013). It is also used to discriminate specific traffic from other types of traffic to protect the network from attack. Therefore, researchers have focused specifically on traffic classification recently, in order to provide a service for other researchers to facilitate the process of intrusion detection.

Different applications are freely available nowadays, which makes the process of communication between people very easy. Users can use an application such as PayPal to carry out financial transactions via the Internet even if living in different countries, and YouTube is heavily used to create own channel on which to post and watch videos, as well as other VOIP applications to find different ways of communicating.

In summary, the Internet network provides many different services for users, making the world like a small village. Traffic classification is a very important part of this process, primarily to deal with security issues concerning the traffic that crosses the network. Classification may be used to detect process to know which packets of data are normal and which are not normal (malicious).

1.2 Thesis Goal

Many people spend a very long time using the Internet daily, depending on their requirements, and making use of various websites and different types and areas of applications. However, security concerns have also increased considerably over the years. Therefore, different protection methods have to be developed to protect the traffic of the data that crosses the network daily. Researchers need to focus on those applications the most used between people recently. Therefore, in this thesis, we focus on some well-known applications in order to distinguish this traffic from other, unidentifiable traffic. This work represents services for researchers that conduct searches in the security domain.

The following applications are those that are used most in contemporary times: Skype[1], YouTube[2], Gtalk[3], Yahoo Messenger[4] and

[1] http://www.skype.com/en/download-skype/skype-for-windows/
[2] https://www.youtube.com/?gl=JO
[3] http://google-talk.en.softonic.com/
[4] http://yahoo-messenger.en.softonic.com/download

PayPal[5]. Therefore, we must classify this type of traffic in order to protect it from any type of attack. In this thesis we generate a new dataset and tested it using four ML techniques on the five different applications together, as mentioned above. This is apart from the VOIP and Non-VOIP applications that we classified in the Non-VOIP and VOIP case and Multiclasses classifications case, in order to discover the differences between them. We also extracted the specific traffic for specific applications, and distinguished this from other, unidentifiable traffic.

We used the Wireshark[6] sniffing tool to capture the application traffic that passes over the network and collect their data. The dataset that we generated includes five different applications together. Collecting from the TL layer that provides data integrity and privacy. Collection was carried out according to the different protocols, including: Transport Control Protocol and User Datagram Protocol (TCP and UDP), Secure Sockets Layer (SSL) and Hypertext Transfer Protocol (HTTP). We also used Waikato Environment for Knowledge Analysis tool (Weka)[7] to classify traffic using four ML techniques that each achieved a high level of accuracy. These algorithms were tested in the dataset in Non-VOIP and VOIP case and Multiclasses classifications case, as well as using four important statistical features to collect different data types and identifying each one. Statistical features from classification methods were used because traditional methods such as port numbers and Deep Packet Inspection (DPI) fail to accomplish classification of encrypted VoIP applications (Alshammari & Zincir-Heywood, 2015). Data for all applications were collected and applied to a live network in real time and real data were collected from the network using an experimental testbed.

1.3 Thesis Contribution

In this thesis, traffic classification for five important applications together has been studied and classified according to two main cases, including: Non-VOIP and VOIP applications case. The second case represented the applications in Multiclasses, and these included PayPal, YouTube, Gtalk, Yahoo Messenger, and Skype. The contribution of this thesis is appeared as follows:

1- A new real dataset was generated from a live network using an experimental testbed that collected data for each application.
2- Traffic classification was carried out for five different applications together in Multiclasses case including PayPal, YouTube, Gtalk, Yahoo Messenger, and Skype, in addition to providing special service for developers to study and analyze the information so as to

[5] https://www.paypal.com/jo/webapps/mpp/home
[6] https://www.wireshark.org/download.html
[7] http://www.cs.waikato.ac.nz/ml/weka/

develop different methods that may be used for protection of such applications from any attack.

3- To represent the importance of the most well-known applications that are used by people in two main parts, namely: VOIP and Non-VOIP that were collected from the TL.
4- Four different statistical features were chosen from many other features and used to support the accuracy of the performance.
5- The dataset was tested using four ML techniques, namely: meta.Adaboost(j48), Random Forest, J48 and MLP classifiers. Traffic in two classification cases Non-VOIP and VOIP case and Multiclasses case was detected and classified.

1.4 Thesis Organization

This thesis is organized as follows: in Chapter Two, the background to certain issues regarding traffic classification is presented and some ML algorithms used for traffic classification are discussed, along with an explanation of how these are used to detect and classify each application. Specific important applications that were classified using ML algorithms are also described and explained.

In Chapter Three, design and methodology are described, and the five main applications whose traffic was detected and classified are presented along with description of the TL layer. Four common ML classifiers are also discussed, as follows: (meta.Adaboost (j48), Random Forest, J48 and MLP). Use of the WEKA toolbox is described and the evaluation criteria/ metrics for implementing ML classifiers are explained. The structures of the generated dataset is described, along with an explanation of the method used to compile it, followed by an overview of the Wireshark sniffing tool and an explanation of how it works.

The practical experiments are presented in Chapter Four, and an explanation of how they were conducted. The results for each ML classifier are discussed, and then confusion matrices for the selected classifiers are highlighted. Finally, the conclusions are presented, along with suggestions for some future research work.

CHAPTER TWO
LITERATURE REVIEW

2.1 Some Issues Regarding Traffic Classification

ML algorithms are capable of providing information for identifying complex or encrypted traffic, like many protection methods such as firewalls. However, they face challenges in classifying the applications in practice. Therefore, in (Shao, Zhang, Chen, & Xue, 2014), "Towards time-varying classification based on traffic pattern", they proposed a model is proposed called a time-varying Logistic Regression model, which is linked dramatically with the traffic pattern. When a comparison is made between the time varying and original Logistic Regression model, there appears to be a clear improvement in accuracy. Therefore, it was important to look at the particular properties when making changes and those that were taking place in the traffic movement in the time domain. In these researchers' work however, they used port number, whereas in our work we used some statistical features that were more dynamic for classifying the packets.

Firewall Protection methods continue to evolve in different aspects. In (Masud, Mustafa, & Trabelsi, 2014), "A data driven firewall for faster packet filtering", they suggested a new technique that works on the basis of data mining and packet filtering. Thus, this was a technical add-on to previous work, based on packet filtering in a set of rules of filtering, where this traditional approach works on the examination of packets by scanning for all rules in the group, until it is required to get mismatches. Since this method is not effective if the number of rules is very large, they had to create an additional technique.

This technique takes every norm of packet filtering as a class alone. Thus, that trains the data and the development of a label contains the packet headers, where they are trained on the existence of the base in matching packet filtering rules, for any packet six times faster than the traditional firewall network. In fact, the effectiveness of this new approach has been proved experimentally and theoretically by means of a real network. However, they had to filter the packet whether or not it was a match or a mismatch without returning to traditional methods.

Applications are still being developed in increasing numbers for various reasons, as well as the increases in bandwidth. On addition to a greater interest in security issues and protection of the information that passes through the networks. Recently, because of all these reasons, traffic classification has become the most important process on which to focus. There are many techniques and issues are related to the traffic classification process. In (Xue, Wang, & Zhang, 2013), "Traffic classification: issues and challenges", they analyzed the techniques currently being used and presented the general challenges with which traffic classification is faced,

such as: Payload-based, Flow-based, and Host-based, as well as other general challenges. Additionally, they presented techniques for each challenge, and also outlined some recommendations; for example: Parallel Classification, Hierarchical Classification, Finding Payload Signatures Automatically, Traffic Analysis Approaches for Flow-based Techniques and Developing Suitable Cleverer Learning Algorithms. However, many issues and difficulties still exist that must be resolved in conjunction with development of applications and increase in traffic over the Internet.

Many research works discuss the classification issues regarding many applications on the network. Different types of ML techniques exist to deal with classification. It is uses the nearest-neighbor (NN) ML algorithm to distinguish the performance of the classification process and this technique has several advantages, one of which is that there is no need for training. Another advantage is being able to deal with a large number of classes, and there is no risk of parameters from the over-fitting problem. In (Zhang, Xiang, Wang, Zhou, Xiang, & Guan, 2013), "Network traffic classification using correlation information", they suggested a new, non-parametric approach to improve the performance of this algorithm in traffic classification. Therefore, this approach takes into consideration the theoretical and experimental aspects and analyzes the information that links the performance to each other. After experimenting with both, these authors suggest three new methods of classification, namely: AVG-NN, MIN-NN, and MVT-NN. They concluded that it is possible that the performance of traffic classification improves significantly and continuously after many experiments. They worked on two sets of traffic data even though the circumstances in which the results were obtained were difficult, such as the lack of training data. This approach could also be used in many different applications. However, they only focused on the nearest-neighbor (NN) ML algorithm.

Traffic classification is a science in itself, and is still in constant development. This science faces many challenges throughout the network. It should be secure, protected and reliable in exchange for the amount of information that passes through it. In (Dainotti, Pescape, & Claffy, 2012), "Issues and future directions in traffic classification", they review and discuss future directions in traffic classification, as well as some achievements that have been made in the previous time period. They also made a comparison with three other main points: Privacy, Reliability and Application. These posed many challenges that were confronted in previous decades, and suggest several strategies to overcome such challenges in order to improve the work on this science and increase its capacity and effectiveness in traffic classification for the future. In this current work, some of these important recommendations are mentioned.

Traffic classification has increasingly proved its importance with the rapid development of applications used over the Internet. It also has a very important role in many aspects, such as intrusion detection and quality of service. In (Zhao,Yu, Chen, Jing, Peng, & Liu, 2012), "A novel online traffic classification method based on few packets", they made a comparison between four methods for traffic classification, namely: Classification Based on the First Few Packets of a flow (CFFP), Classification based on the Entire Packets of a flow (CEP), Classification based on Arbitrary Disjunctive Few Packets of a flow (CADFP) and Classification based on Arbitrary Conjoint Few Packets of a flow (CACFP). Their results show, without using port features, that the highest accuracy of classification was CADFP and CACFP as compared with the others. The CADFP and CACFP methods are more efficient and effective in strengthening the classification process, especially for online traffic. The solution they found in the first part of the packets is by random selection, including analysis of the result of classification on two datasets. Furthermore, these two methods faced some challenges, so there is no method without drawbacks.

Security are the most important characteristics when exchanging information via the Internet. To support previous properties, security and safety are based on filtering and classifying Ethernet packets within network devices, such as intrusion detection, routers and firewall systems. In (Wicaksana & Sasongko, 2011), "Fast and reconfigurable packet classification engine in FPGA-based firewall", they presented fast architecture and a reconfigurable Packet Classification Engine (PCE). This engine in the firewall was based on the FPGA that depends on a tree algorithm. It also inspects the multi-dimensional field of the packet header.

This algorithm leads to simplifying the system and making it safer. It is based on destination IP Address, Source Port, Source IP Address, Destination Port and Protocol fields of the packet header. The PCE examines the Ethernet packet to identify which of these packets are normal and which are dangerous before investigating the content. The PCE is not yet complete, with a number of aspects still to be explored, such as the rule update mechanism. It is still using the traditional ways, like port numbers, to classify the packets.

In (Li, Claypool, & Kinicki, 2015), "Treatment-based traffic classification for residential wireless networks", they used the NS-2 simulator to produce Classification And Treatment iN an Access Point (CATNAP) when more than one application are running simultaneously. They addressed in automatic form the flows that pass through the access points in the wireless network without any user interaction by means of simulating different situations and used three methods of clustering, namely: CATNAP, DropTail and SPQ, and applied data to them. It became

clear that the CATNAP performance was better than either the DropTail or the SPQ because exiting was shortened in network performance for different types of latency over long, medium or short periods of time. Therefore, the simulation also showed that using (CATNAP) improves the quality of service performance under a wide range of network conditions. In our work we use real data from a live network and not simulating tools.

2.2 Classifying General VOIP and Non-VOIP Applications

It is difficult to determine intrusion detection and protection by using application identification through the network because the applications, bandwidth and the large amount of packets is growing continuously. In (Qin, Wang, Liu, & Guan, 2015), "Robust application identification methods for Peer to Peer (P2P) and VoIP traffic classification in backbone networks", they found a solution for the amount of packets to reach the goal of P2P and discrimination applications through (VOIP). Thus, they employed the Bi-flow model to collect traffic packets in order to extract the characteristics of mutual behavior through the various terminals. To capture the flow dynamically, the Packet Size Distribution (PSD) is used. This expresses the probability distribution of the length for the payload of the packets in one Bi-flow.

The next step was to collect the previous feature (PSD) information in different applications for each P2P and VOIP. The results of the analysis proved that the results of this step using (PSD) vary from one application to another. Therefore, this difference can be used to identify the traffic. A new robust traffic identification method was based on PSD, and was only concerned with the whole connection time in the first few packets. In addition, the existence of the stable elements from the base on the whole connection time in the network was employed on the first few packets to capture the dynamics of the flow.

To reduce the data and to make for easier handling, they used a method called Poisson sampling. The experimental results based on the effects of traffic that was collected from the university platform showed 97% accuracy of the proposed method and it is therefore a strong technique that can be used on traffic control in real-time. However, they focused on only one feature, called PSD that did not support this work completely. In our work we used four different features to support classification of five classes which enhanced the results.

In (Alshammari, & Zincir-Heywood, 2015), "Identification of VoIP encrypted traffic using Machine Learning approach", they focused on VOIP applications. Furthermore, they used several methods to recognize and classify the encrypted traffic flow to generate robust signatures for identifying the encrypted traffic. They used three different ML algorithms, namely: Adaboost, C5.0 and Genetic programming (GP). They also applied

a statistical calculation to a network flow in order to extract a set of unique features for each application, in addition to applying to many types of datasets for testing and training. According to the testing and the result, the C5.0 algorithm was preferred. In contrast, in our work different ML classifiers were used, and meta.Adaboost (j48) achieved the highest rate for accuracy.

Uses for the VOIP applications continue to increase extensively. With this continued popularity, there is also an increase in security concerns that revolve around these applications. In (Sinam, Singh, Lamabam, Devi, & Nandi, 2014), "A technique for classification of VoIP flows in UDP media streams using VoIP Signalling traffic", they proposed a method to detect communications via the Internet, specializing in passage flows through UDP media streams.

They were particularly interested in detecting the traffic passing through the Skype application. This interest was based on heuristics to identify the RTP or RTCP in the UDP packet header without using payload information. Using the Skype-signal and Skype-media, the heuristic was based on the Start of Message (SoM). Their results were validated by using more information in the behavior of the host. However, this work depended on one VOIP application called Skype. Our work was built around five different applications separated into VOIP and Non-VOIP applications.

The number of VOIP applications and the number of algorithms and techniques used to track the traffic and detect these applications have increased considerably in recent years. In (Fonseca, Cruz, Simoes, Monteiro, Silva, Gomes, & Centeio, 2014), "A comparison of classification techniques for detection of VoIP traffic", they studied the techniques used in the detection of traffic in (VOIP) applications. They worked on two major categories: profiling of network traffic patterns and modeling of communication flows for anomaly detection. They discussed many techniques and algorithms in those categories. This work confirms that the legacy ways depending on the port number and protocol in detection and tracking, accuracy are less than modern ways. Modern ways are divided into two categories, the first of which does not require any traditional information such as protocol and port number; there are also differences between the algorithms in this category, in that they are exclusively dedicated to a particular type of VOIP application. The second category specializes in creating the models for the channel of communication service flow in the detection process.

The firewall device specializes in protecting traffic and preventing those that are unwanted, using different filtering policies. In (Duan, & Al-Shaer, 2013), "Traffic-aware dynamic firewall policy management: techniques and applications", they describe, classify, and compare traffic-aware firewall policy management techniques. Their work is based on

some important points, such as aims, complexity, schemes, limitations, and applicability. They adopted the classification process for traffic-aware firewall policy techniques in two dependent categories, namely: matching optimization and early rejection optimization schemes.

The first category contains technology that reduces matching time in normal network traffic. The second category is a technique aimed at reducing the size of the rules or conditions on the traffic; possibly to filter as much unwanted traffic as possible. These two categories are dynamic and self-adaptive to ensure good performance gain for the network. This helps engineers and researchers in the process of understanding and solving the problem. They also adopted appropriate techniques based on application requirements, although they focused on only two categories regarding policy techniques.

It has become necessary to perform Traffic Classification for several reasons, including the rapid development of applications and protocols used by the Internet. In (Tapaswi & Gupta, 2013), "Flow-based P2P network traffic classification using Machine Learning", they used an estimator called Naïve Bayes to classify traffic based on features of P2P networks, where this network has the largest volume of bandwidths. This estimator classifies traffic into P2P and non-P2P networks. The results achieved produced a high level of accuracy and this appears to be the case when using a good training dataset and when the correct features are obtained. A high level of accuracy can be achieved from simple Bayesian algorithms; this algorithm produced a level of accuracy ranging from 65% to 85%. The amount of data used also plays a very important role in determining the accuracy of the algorithm according to some studies, although other classifiers can achieve an accuracy rate higher than the previous ones. This is the same as in our work when using the meta.Adaboost (j48) classifiers which achieved an accuracy rate of 98.3007 %.

In (Ibrahim, Nor, Mohammed, & Mohammed, 2012), "Taxonomy of Machine Learning algorithms to classify real time interactive applications", they discuss about classifying two interactive applications, namely: online TV and Skype. The interactive application has become wide-spread and important for people in the last few years. Therefore, focusing on it is very important. These authors used the Wireshark tool to capture the packets that were transmitted over the network.

The measurements were based on two features: interval time and packet length. They selected these features for ML in order to reduce the complexity of classification, particularly when dealing with a real-time application that is very sensitive. After collecting the captured file, it was separated into two parts: the training data and testing data. These were submitted to the WEKA tool which compared ten different MLs and extracted the results. The Random Forest algorithm gave a high accuracy

result of 99.8%. However, they used only two features on two applications. In our work, we used four features on five applications, including interactive and non-interactive applications that get more contribution.

Many disadvantages became apparent in the classical methods that were used in traffic classification, such as the port number, the payload information and the encryption technology to avoid detection. To avoid some of these disadvantages, other researchers suggested recently using ML methods, including supervised learning and unsupervised learning. However, both methods have their disadvantages, the first of them is the problem of dealing with labeled instances, and the second is the problem of the long time it takes to work in the case of a manual state. To solve such problems, in (Mahajan & Verma, 2012), "Implementation of network traffic classifier using semi supervised machine learning approach", they proposed a new technique called Semi Supervised ML. This technique creates a classifier from a training dataset, consisting of both labeled and unlabelled instances. They used a MATLAB tool to evaluate and compare the different performances of the classifier, based on three different ratios of the labeled instances in the training dataset. The result showed that the classifier had the best performance at 30% of the cases classified as labeled instances, in a training dataset in the number of clusters equal to 50. The accuracy of the classification was up by 94.7%. However, there are drawbacks, the requirements depending on the work that was researched. It is possible to achieve better results in other methods, as in our work that used a supervised ML to achieve 98.3007 % accuracy.

Monitoring the Quality of Service (QoS) in the network is very important. In (Bujlow, Riaz, & Pedersen, 2012),"A method for classification of network traffic based on C5.0 Machine Learning Algorithm", they carried out an analysis, especially in Multi-hop networks. This requires knowledge of the information about traffic and the types of applications that are being used over the network to accomplish the task. To overcome the defects in the existing methods of traffic classification, their work suggests a new method of ML algorithms called C5.0. This is based on the statistical information received from the traffic algorithm that is applied by C5.0 which shows outgrowth for this algorithm. They were able to distinguish seven different applications in a test set of 76,632–1,622,710; the average accuracy of the unknown cases was 99.3–99.9 %. They used high-quality training data collected by their system.

This algorithm was obtained with high precision by using a unique set of criteria for both training and classification information. The different applications which classify these are interactive application such as Skype, Games and SSH. Classification of the traffic appears to be similar to radio streams via a web page and web browser traffic. It can also classify the FTP and torrent. The tests have still to introduce improvements to this

proposed approach. However, it is facing difficulties in classifying FTP and torrent, because the characteristics of the flows are very similar.

2.3 Some Other Specific Classified Applications

In (Alshammari & Zincir-Heywood, 2011), "Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?", they depended on presenting an ML algorithm that employs a set of statistical features and simple packet header feature sets without using source/destination ports, IP addresses and payload information to describe the encrypted application tunnels in the network traffic. This approach uses an analysis tool that is applied to two encrypted applications, namely Skype and secure shell (SSH), using different traces from different networks. The final result refers to the possibility of identifying tunnels of the encrypted traffic with high accuracy. There is also the possibility of identifying the services that run in the encrypted tunnels without including the above three pieces of information. The GP algorithm achieved 89% (Detection Rate) DR and 0.2% (False Positive Rate) FPR in the test performance when it trained on one network but was tested on another. In contrast, our work classifies five applications, namely: Skype, Gtalk, Yahoo Messenger, YouTube and PayPal.

Skype has become the most well-known application for communication between people and is commonly used for this purpose because it provides several services, including voice communication, communication via video, sending files and chat. Because of its importance it was necessary to suggest methods/algorithms to detect and classify Skype traffic. In (Adami, Callegari, Giordano, Pagano, & Pepe, 2012), "Skype-Hunter: A real-time system for the detection and classification of Skype traffic", they proposed a real-time algorithm called Skype-Hunter to classify and detect Skype traffic.

This algorithm uses the signature-based and statistical procedures which are used to enable the classification of data traffic signals, data traffic of calls and data transfer. This algorithm was applied to many datasets that were collected from different network scenarios. The system used here outweighed the classical statistical methods of traffic classification. The analysis of performance showed very good results for the different types of traffic traces in different access networks. However, in our work, we studied applications not just for Skype and these provide the same services nearly as Skype, such as Yahoo Messenger and Gtalk.

Many VOIP applications on the Internet are used increasingly and are becoming more popular day by day. Skype is the application that is used the most, and so it is difficult to find and classify data relating to it in the general. Because of this, Skype uses different encryption mechanisms, as well as following the proprietary design and a closed source. Many

methods and algorithms have evolved to carry out the process of traffic classification with high accuracy results.

However, these proposed methods require a great deal of computing resources, especially with the speed of existing networks. Therefore, in (Del Río, Ramos, García-Dorado, Aracil, & Cutanda-Rodríguez, 2011) "On the processing time for detection of Skype traffic", they focused on minimizing the cost of processing algorithms used in detecting Skype traffic. Using algorithms which were applied and validated previously, the information from Network Interface Card (NIC) and memory Consecutive was able to be read, working on 1 Gbps and 3.7. The percentage of (FN) was 6% in the worst case, where the (FP) rate equalled zero. This approach was also applied to a P2P network and with the detection technique DPI. These authors are still continuing to research different areas in this regard.

CHAPTER THREE
DESIGN AND METHODOLOGY

3.1 Information about Applications and Their Layer

Many applications have been developed and come into use recently. The most used are VOIP applications which have numerous uses for every type of user. The VOIP is a technology that allows users to communicate with each other over the Internet protocol. This category includes different applications including Skype, Gtalk and Yahoo Messenger. These applications also use the TCP and UDP protocols from transport layers. They are much sharper than traditional telephone networks (Ibrahim, Nor, Mohammed, & Mohammed, 2012). In addition, Non-VOIP application can also facilitate some things such as watching videos on the YouTube site, establishing special work by creating one's own channels and managing financial transactions online using the PayPal website. In the following we shall explain the above-mentioned applications in detail.

Skype is the most well-known application used by people to communicate face to face. It is also the largest VOIP application that uses different types of P2P network. It is the best VOIP application with a high quality of sound (Adami, Callegari, Giordano, Pagano, & Pepe, 2012) and in video and voice calls provides many safety settings for users. It can be installed on different devices including Ipads, smart phones and personal computers, etc. The Skype application enables video calls, voice calls, and text chat and it can be used to send text files and pictures.

The Yahoo Messenger application is called an all-in-one communication tool. Anyone using Yahoo Messenger can benefit from different services such as email, voice calls, video calls, SMS messages and sharing photos with family and friends. It can also be set at different settings for privacy to protect personal information, as well as changing online status according to the user's needs. Chat rooms of various categories are another important feature.

Many people use a Google site for various purposes, such as using the Google search engine to access a wide range of topics, as well as creating an email account (Gmail or Googlemail) which includes not only email services but also the capability of attaching different types of files, etc. Another important service from Google is Google talk messenger which includes video calls, voice calls, and text chat. Both Google and Yahoo search engines provide almost the same service as regards email and search, but operates lightly differently depending on what the user requires.

YouTube is the site that is accessed most often for watching videos. It offers the benefit to users of being able to upload one's own work onto the Internet: seeing video recordings online and during live broadcast, sharing videos, writing comments, marking 'like' or 'dislike'. People who

create their own channel on YouTube are called ‘YouTubers’ and this has helped some people to become famous and have their work known globally – something that would not have been possible without such an application. This fame helps him to establish actual work and benefit from it. Different levels and types of security are available on the YouTube site. Users can select any level of security according to their needs. This site also uses Adobe Flash technology to display animation videos and uses different technology for high clarity and more quality. In addition, the YouTube site uses secure Hypertext Transfer Protocol (HTTPs) to provide a secure connection.

Management of financial transactions online can be used by individuals, companies and venture capitalists and the process has become very easy. Users of PayPal site can create an account to benefit from all the services provided as long as they have a credit card such as MasterCard or Visa card, etc to gain access to the site. Different services are provided, including sending payments, requesting payments and withdrawing money. This site is very safe as it gives the user a choice of many security levels associated with various main topics, such as personal information, account information, money information and all material information. It also uses SSL protocol to provide a highly secure level of protection for its services. This site provides a wide range of services for managing money via the Internet, obviating the need to physically go to a bank and conduct business in the traditional way.

The previously mentioned five applications were chosen from the TL. The TL is responsible for delivery of the message between one process and another, the process representing the application program that is running in a host. It ensures that the entire message arrives in order and correctly. The TLS protocol is similar to the protocol that is called SSL and it contains many versions of it. These protocols provide a high level of security and safety for communicating on the Internet and this is also called a cryptographic protocol. It is designed as a secure communication channel between the client and the server.

The way of communicating is designed to prevent tampering, eavesdropping, or message forgery. In addition, this layer is created to produce security services and data integrity for the communication channel over a reliable transport protocol such as Transport Control Protocol in the transport layers. It also uses different methods to encrypt the data that crosses the communication channel. Various protocols are used in these layers, including TCP, UDP, HTTP and SSL protocols that were used in our five applications.

TCP is a byte-oriented protocol for storing messages received from the process as a stream of bytes and sends them in segments. The TCP is also a reliable protocol because it provides important services including

detecting duplicate segments, replacing lost segments, and the end process delivers the bytes in order, etc. UDP protocol is a message-oriented protocol, which means that the process delivers a message to the UDP protocol. This message encapsulated in a user datagram and is sent over the network. Each message is separate and independent of any other message that is sent over the network. It is considered as a feature when using an application such as transmission of real-time data.

However, UDP protocol is unreliable in that the sender does not know the destination of the message that was sent (Behrouz A.Forouzan, 2004). HTTP is the main protocol that is used on the World Wide Web (WWW).It is responsible for the messages that are formatted and transmitted and determines what action the web server and browser should take when a command is received and responds. HTTPs protocol is also called HTTP over TLS (Needleman, 2000) and provides secure communication over the network. It also provides communication over the HTTP with a connection encrypted by TLS.

3.2 Machine Learning Classifiers

We are evaluate four different supervised ML algorithms on our generated dataset. Supervised learning uses labeled training data that makes predictions based on evidence in the presence of uncertainty, to conclude a function. The percentages of accuracy vary between the meta.Adaboost(j48), Random Forest, J48 and MLP classifiers. In addition, there are also differences between the two cases of classification here: Non-VOIP and VOIP case and Multiclasses case for all five applications represented in five classes (Skype, Gtalk, Yahoo Messenger, YouTube and PayPal). In the following sections we shall explain ML classifiers in detail.

3.2.1 meta.Adaboost (j48):

This is an ML algorithm shorted for Adaptive Boosting that was formulated by Yoav Freund and Robert Schapire in 2003 (Alshammari & Zincir-Heywood, 2011). Boosting is a small band method that originates from the main classifier and is prepared from training data. The second classifier was created to focus on those instances of the training data when the first classifier obtained an incorrect result. The process of adding classifiers continues until maximum accuracy is achieved. It usually improves the Performance significantly, and is also adaptive, which is significant for the instances that are misclassified from previous classifiers (Tiwari & Prakash, 2014). In addition to improving performance, this can be used concurrently with many other types of learning algorithms. The output of the other weak learning algorithms was collected to the weighted sum that represents the final output of the boosted classifier.

Strong classifiers are created from a linear combination of weak or simple other classifiers. A strong classifier can also be used to enhance the result according to the user's needs. Thus, the boosting process was used here to enhance the performance of the J48 decision tree ML algorithm. The accuracy results for each J48 only and for the meta.Adaboost (j48) ML algorithms represent the enhancement of the performance that was produced on the J48 decision tree after the boosting process for Non-VOIP and VOIP case and Multiclasses classification case. For our work, meta.Adaboost(j48) proved to have high efficiency in classification after achieving the highest accuracy level among all the other classifiers.

3.2.2 Random Forest:

ML algorithms were developed by LEO Breiman and Adele Cutler. Random Forest is a supervised ML technique based on the collection of a large number of decision trees. It also selects features randomly in order to obtain an individual tree and it runs efficiently on large databases. A single forest contains a number of trees and each forest refers to a prediction class for new unlabeled data. In addition, it is an effective method for estimating the amount of missing data and maintains accuracy when a large proportion of data is missing. The depth of the tree can be determined by the node size parameter. In addition, this prediction is made by aggregating, that is, the majority vote for classifying the predictions of the ensemble.

The forest is built in order to classify the process for a new instance. It is run among all the trees that are constructed in the forest. Many trees will be generated by the classification process. Each individual tree is classified as a new instance and given a vote, and then all the votes from all the trees are combined. The class that achieved the maximum amount of votes – called majority voting – is declared as the classification of the new instance. The Random Forest classifier is evaluated based on its accuracy and error rates by applying this classifier on the dataset.

To measure these rates two ways of compiling training and testing data were used, including dividing the dataset into a percentage for training and testing, and repeating the data using 10s folds or 20s folds iterations, etc. The effectiveness of the traditional random forest classifier can be high. On the other hand, there can be side effects for this classifier, the main one being the usage of memory that is called memory bound (Van Essen, Macaraeg, Gokhale, & Prenger, 2012).

Figure 3.1 shows a section of random forest representing a maximum tree depth of 3-6 for decision trees.

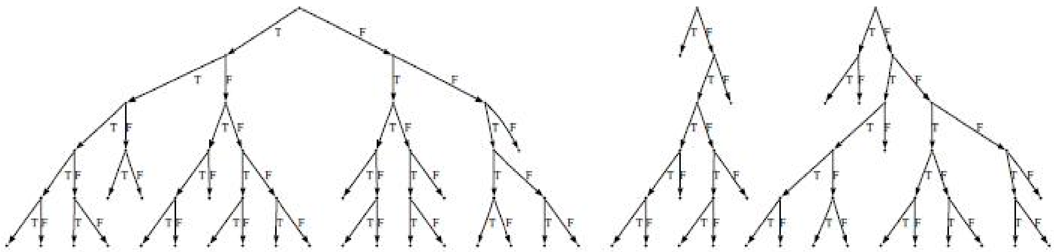


Figure 3.1
Part of Random Forest of Decision Tree.
(Van Essen, Macaraeg, Gokhale, & Prenger, 2012)

3.2.3 J48:

This is also called C4.5 and it is a binary decision tree based on a classification algorithm. It is a popular early ML method. In addition, it is implemented based on a divide-and-conquer strategy and is represented in an hierarchical data structure form (Alshammari & Zincir-Heywood, 2011). It is a top-down induction of decision trees and is based soundly on information theory, as a means of knowing which attribute to select. It represents trees that are easily understood by users. It is also capable of stopping the splitting process when the number of nodes is very small, and the default value for it equals two nodes. It can look messy and complicated when the decision tree is of a large size. In addition, the decision is grown using a strategy called Depth-first. After splitting the dataset, the best information that was achieved in the test is selected from other information (Zhao & Zhang, 2008).

In this case, the decision tree was used for the classification process. In addition, it can quantify the goodness of a split for the decision tree using the impurity measure. For all branches together their split is pure, and after the split, all instances of choosing a branch belong to the same class. Entropy is a possible function to measure the impurity of the split of the decision tree. Equation 1 accounts for the entropy measure. There may be instances when there should be a split to decrease impurity if the splits are not pure. Therefore, to calculate the total impurity and measure it we use Equation 2 (Alshammari & Zincir-Heywood, 2011).

$$J_m = -\sum_{j=1}^{n} \left(p_m^i \log_2 p_m^i \right) \quad (1)$$

$$J'_m = -\sum_{j=1}^{n} \frac{N_{mj}}{N_m} \sum_{\mathbf{i=1}}^{\mathbf{k}} p_{mj}^i \log p_{mj}^i \quad (2)$$

Where $_m$ it's a node, P_m is the probability of $_m$, J48 was eventually developed by Ross Quinlan. It is also an open source Java implementation of the C4.5 algorithm in the WEKA classification tool. C4.5 is a program that a decision tree creates based on a set of labeled input data.

3.2.4 MultiLayer Perceptron (MPL):

This is a feedforwared artificial neural network ML algorithm. It is the most commonly used algorithm of all other types of neural networks. It shows a model map set of input data to give a set of appropriate outputs. It can contain many input layers, one or more hidden layers being in the middle and many output layers that contain computational nodes, as shown in Figure 3.2. An MLP algorithm is a directed graph that consists of multiple layers of nodes, each layer making a full connection to the next one, except the input nodes. In addition, it is a modification of the standard linear perceptron and can discriminate between the data that are not linearly separable. Input signals are transmitted over the network in a forward direction, and they cross layer by layer in the MLP (Kevric & Subasi, 2012).

One of the most important features of the MultiLayer Perceptron (MPL) is its ability to create a model for smoothing any functional relationship, and can be between one or more predictors and the irrelevant weights (Abderrahim, Chellali, & Hamou, 2015). In the non-linear activation function each node is called a neuron or processing element. Each neuron has a value that is calculated from the weighted values of the previous input neurons and is summed with the input values, individually for each neuron, plus the bias term. It can be seen that the MLP classifier achieved the lowest accuracy result. However, the result for the Non-VOIP and VOIP classification case is better than that for the Multiclasses classification case.

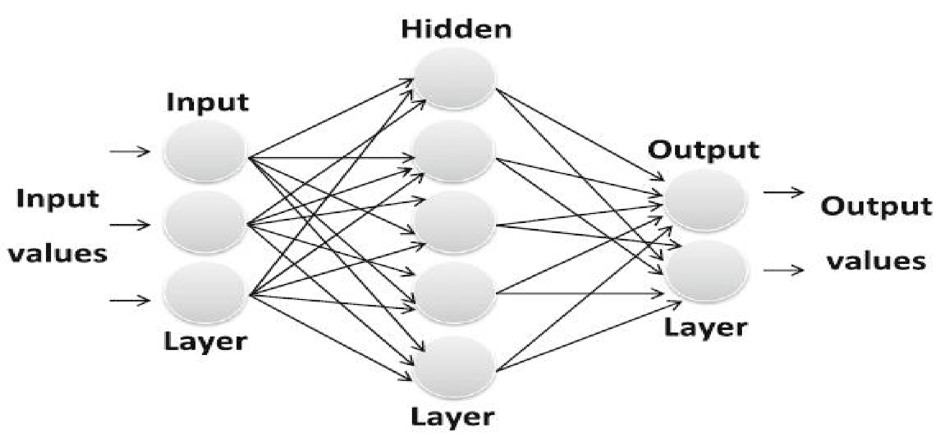


Figure 3.2
MultiLayer Perceptron (MPL) Structure.
(Abderrahim, Chellali, & Hamou, 2015)

We can calculate the hidden layer by using the number of input neurons represented in I and a set of weights assigned to them between the input and hidden neurons represented in wij. Equation 3 shows how to calculate the outputs of all the neurons inside the hidden layer. Equation 4 shows the results for the output layers and Equation 5 represents the sigmoid function which was used in more than one hidden layers.

$$O_i = \sum_{i=1}^{N} wij\ \psi i \qquad (3)$$

$$Y^{\wedge} = f\left(\sum_{J=0}^{\prime\ m} wJ\ \ y_J^H\right) \qquad (4)$$

$$Z(x) = \frac{1}{1+e^{-x}} \qquad (5)$$

The symbols in the previous equation refer to: i = 1,2...,N and j=1,2...,M and z, Yj are the activation functions, z represents the sigmoid function and j^{th} nodes represent the hidden layers. The activation function for the output layer is represented in f as a linear function. Its represents the output of the neural network obtained from a single neuron in the output layer.

3.3 Waikato Environment for Knowledge Analysis (WEKA)

WEKA is toolbox that was developed at the University of Waikato in New Zealand(Jagtap & G., 2013). It is an open source data mining software suite written in Java language. It is free software that is available for all researchers in this area. Researchers use the WEKA tool to make classifications for different datasets using different types of ML classifiers.

WEKA provides a collection of algorithms and visualization tools with which to analyse data and create predictive models. It also provides interactive graphical user interfaces in order to access the services easily. The group of ML algorithms in the WEKA toolbox can be used to solve the problems of real-world data mining. The data included in the dataset are represented in different formats, such as numeric or nominal attributes and some of other types for the attributes that contains in dataset are also supported. In addition, it supports different standard data mining tasks, but first it is necessary to preprocess the data. Any other tasks may then be carried out according to the user’s requirements, including classification, clustering, association, feature selection and visualization.

A classification task may include many types of supervised ML algorithms, such as: meta (e.g. meta.AdaboostM1, etc), functions (e.g. MLP, etc), trees (e.g. J48 and RandomForest, etc), bayes (e.g. NaiveBayes, etc).Many options may also be available for the test, including cross-validation based on fold numbers for the data or the dataset may be split into training and testing data using a predetermined percentage and other

options. A clustering task contains different types of unsupervised ML algorithms, such as clusterers (e.g. FilteredClusterer).

It is used the same test options like classification but its replace cross-validation to classes to clusters evaluation. Association is a method using in large databases to discover any interesting relationships between variables, such as associations (e.g. Apriori, etc). Feature selection means selecting attributes that can make a mode for it, one of them by selecting all training data or making cross-validation by determining fold and seed numbers. Figure 3.3 shows the interface for the WEKA toolbox, viewing the main tabs that were mentioned previously, we show a part from our work.

Figure 3.4 shows the form of the WEKA toolbox after choosing one of the classifiers from the classification tab and applying it to the dataset, after using the cross-validation from the test options and determining 10 numbers for the folds. In addition, the results were calculated for many of the measures used to evaluate the performance of the classifiers that were used: accuracy rate, precision, recall, Correctly Classified Instances, Incorrectly Classified Instances and many others.

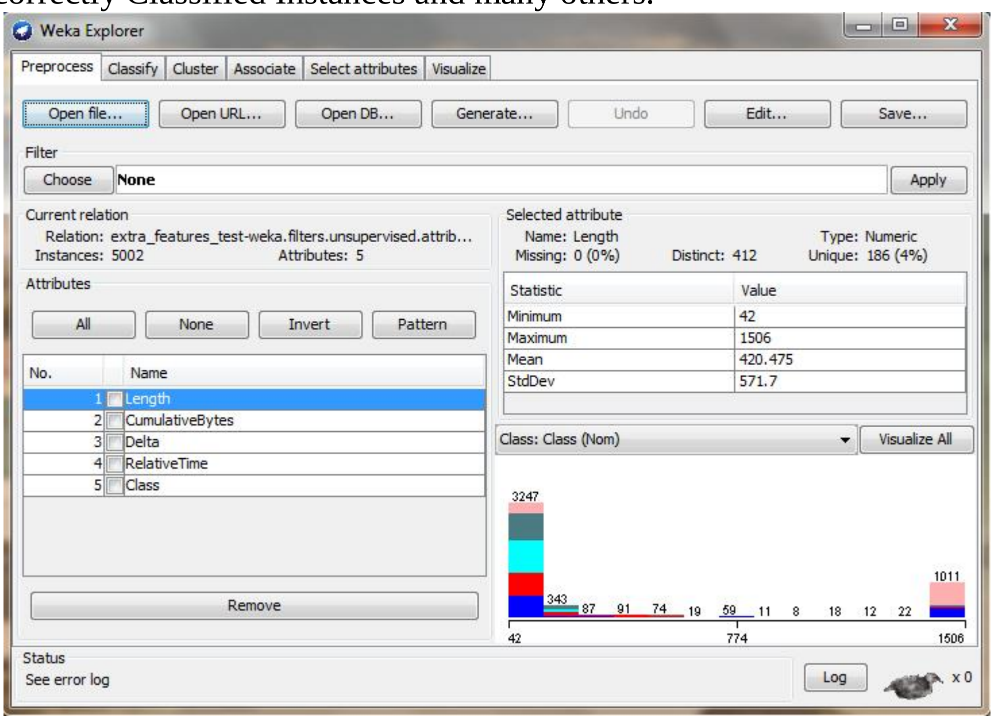


Figure 3.3
WEKA Toolbox Interface.

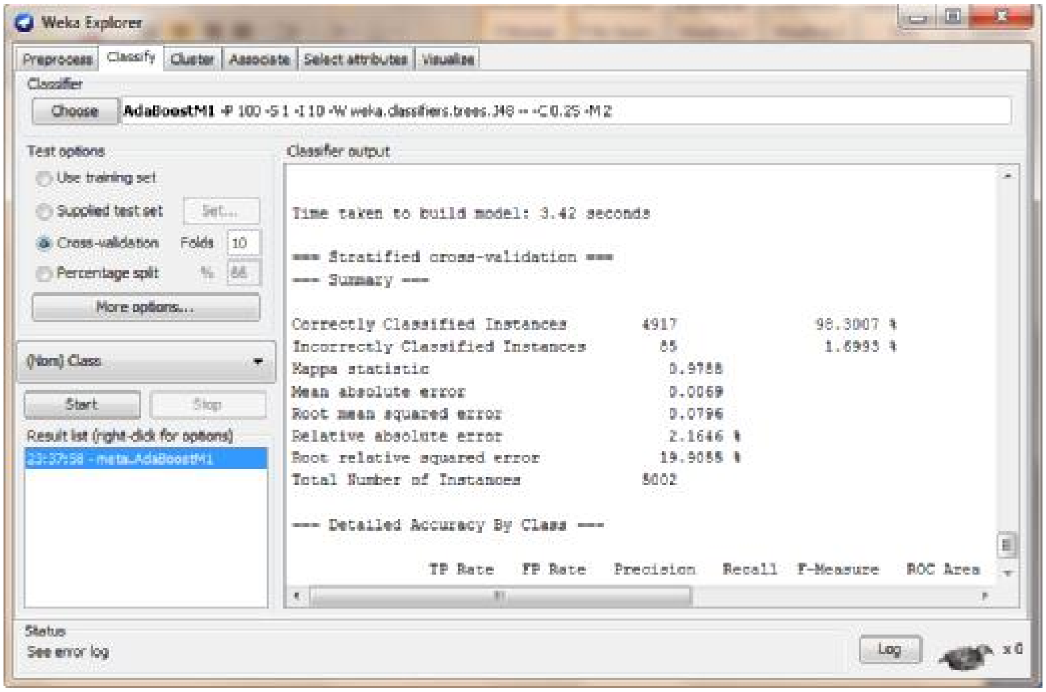


Figure 3.4
WEKA Toolbox Options and Results.

To use the WEKA toolbox the file must be in EXCEL or in Attribute Relation File Format (ARFF) that contains rows and columns. Comma Separated Values (CSV_Editor) helps in converting any file format to the required formats. For example, when using the Wireshark sniffing tool to capture the packets the capturing file can be exported in CSV formats that are easily convertible to ARRF formats. The WEKA toolbox offers many advantages (Jagtap, 2013) to its users, including the following:

1. Its ease of use for graphical user interfaces (GUI).
2. Provides complete collection of data preprocessing and modeling techniques.
3. Portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform, including Windows, UNIX, and Apple Macintosh.
4. Free availability, especially under the GNU General Public License.
5. The user can try and test many classifiers on their dataset, to achieve good accuracy rates and choose from it according to individual needs.

3.4 Evaluation Criteria / Metrics

To prove good performance in the classifiers that were applied to our dataset, we had to use evaluation criteria to measure and explain the classifiers' performance. Basic performance is shown by a confusion matrix. The level of performance appears when the value of the diagonal part is high or low as compared with the upper and lower parts. Therefore, to achieve the best performance the diagonal part must be the best one of

other parts. In Table 3.1 is shown the basic structure for the confusion matrix that contains predictive and real classifications and some other evaluation criteria formulae. We discuss and show the results of the following criteria in detail in the next chapter.

Table 3.1
Confusion Matrix Structure.

	Predicted as Positive	**Predicted as Negative**
Classified as Positive	TP	FN
Classified as Negative	FP	TN

True Positive Rate (TP):

This is a scale to indicate the percentage of instances that are classified correctly in the specific correct class. Equation 6shows how TP is calculated:

$$TP = \frac{TP}{TP+FN} \qquad (6)$$

False Positive Rate (FP):

This is a scale to indicate the percentage of instances that are classified wrongly in the specific wrong class. Equation 7 shows how FP is calculated:

$$FP = \frac{FP}{FP+TN} \qquad (7)$$

True Negative Rate (TN):

This is a scale to indicate the percentage of instances that are classified correctly in the other correct classes. Equation 8shows how TN is calculated:

$$TN = \frac{TN}{FP+TN} \qquad (8)$$

False Negative Rate (FN):

This is a scale to indicate the percentage of instances that are classified wrongly in the other wrong classes. Equation 9 shows how FN is calculated:

$$FN = \frac{FN}{TP+FN} \qquad (9)$$

Average Accuracy Rate (AA):

This is the main criterion with which to measure the performance of the classifiers. It represents the number of instances that were classified correctly for Non-VOIP and VOIP and the Multiclasses classification cases. Equation 10 was applied to obtain the average accuracy rates, as follows:

Average Accuracy (AA) = # correct predictions / # total data points=

$$AA = \frac{TP+TN}{TP+FN+FP+TN} \qquad (10)$$

Recall:

This is the percentage of correct classifications in the case of true positive (TP) out of the instances that were actually positive. This is called positive sensitivity, and can be calculated as in Equation 11:

$$Recall = \frac{TP}{(TP+FN)} \qquad (11)$$

Precision:

This is the percentage of instances that were classified as positive and which were actually positive. This is called positive predictive value, and can be calculated using Equation 12, as follows:

$$Precision = \frac{TN}{(TN + FP)} \qquad (12)$$

F-Measure:

This is a measure to test the accuracy of the performance using both recall and precision. The calculation of the F-Measure is shown in Equation 13:

$$F - Measure = \frac{2TP}{2TP+FP+FN} \qquad (13)$$

Root Mean Squared Error (RMSE):

This is a frequently used measure of the differences between predicted values and real values. If the values are low, that refers to the evaluation that is mostly accurate, and it also reduces the errors according to their values, e.g. if zero value that means there were no errors.

3.5 Dataset Generation

In this section, we provide an overview of our dataset, the tools and classes used, the classifiers that were tested on it and the results of the classification process on the Non-VOIP and VOIP classes and Multiclasses classification (Skype, Gtalk, Yahoo Messenger, YouTube and PayPal). We collected real data for our dataset using the Wireshark sniffing tool for five different applications from a live network. Figure 3.5 shows the block diagram containing the processes for our database generation in detail.

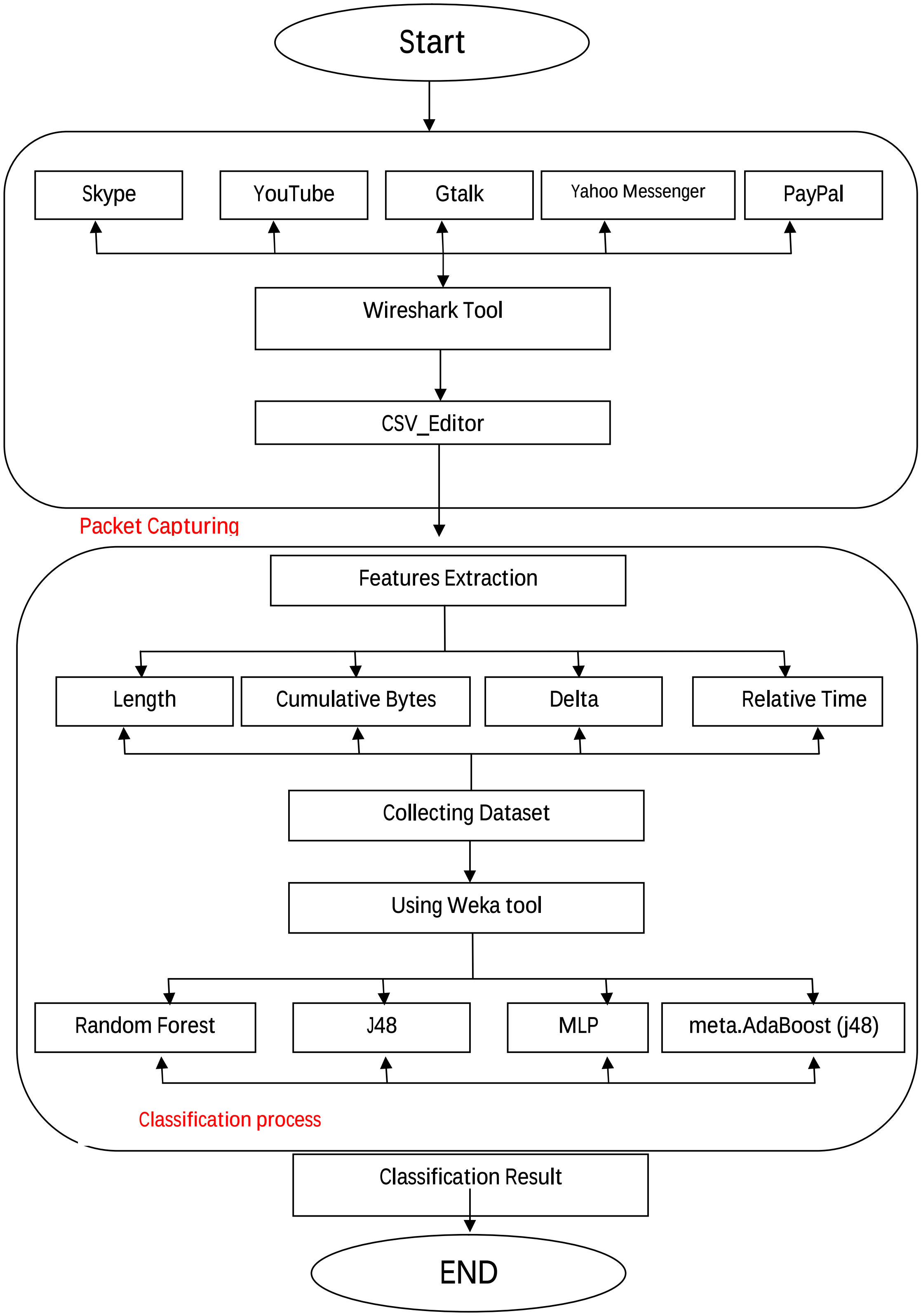


Figure 3.5
Block Diagram for Dataset Generation Processes.

In this section we shall describe the building of the block diagram that explains the structure of this work. The diagram shows the process that

was required to generate our dataset. In addition, no other dataset has been generated to classify the traffic for five important applications in Non-VOIP and VOIP classification case and Multiclasses classification case as well as containing real data from a live network. Our dataset contained 5002 records with four features and two cases of classification, including Non-VOIP and VOIP classes. These include only two classes and each class contains two or three applications. Another case, called Multiclasses, consists of five applications, each one representing a separate class. The classes for each type are shown in Table 3.2 and Table 3.3 for Non-VOIP and VOIP and Multiclasses classification cases respectively.

Table 3.2
Classes for Non-VOIP and VOIP Classification.

Class number	Class name
1	Non-VOIP
2	VOIP

Table 3.3
Classes for Multiclasses Classification.

Class number	Class name
1	YouTube
2	PayPal
3	Skype
4	Gtalk
5	Yahoo Messenger

In the first process in the block diagram we carried out different steps. Firstly, after we start, the Wireshark tool was used to capture the traffic for each of the following applications: YouTube, PayPal, Skype, Gtalk and Yahoo Messenger. Secondly, the capturing file for each application was collected in one capture file. The third step was to edit the files using the Excel program to remove unneeded features, in addition to making the process of converting the capture file from the Wireshark format and exporting it in CSV format to the WEKA tool format (.arff) using CSV_Editor. This was to enable the capture file to enter into the WEKA classification tool later on.

In the next process, the statistical features which extract important information for packets that cross the network were selected. In this work we used statistical classification by using statistical information from other classification methods which use classical and traditional methods like port numbers and DPI. Thus, we selected four important statistical features from many others which are also serving us in this work. These included Packet Length (Length), Cumulative Byte, Delta (Delta time) which represents inter-arrival time and Relative Time features. A description of our statistical features is as follows:

A. Packet Length (Length):

This is one of the most important statistical features used in the traffic classification process. It shows the length of each packet that crosses the real network, as well as calculating the length of the capturing live packet data in real time. The network layer is responsible for the packets and ensures that one packet gets from the source point to its final destination. The original size of the transmitted user data was between 46 and 1500 bytes. The general format for packet length is shown in Table 3.4.

Table 3.4
Packet Size Format.

Preamble	Destination Mac address	Source Mac address	Type/Length	User Data	Frame Check Sequence(FCS)
8 Byte	6 Byte	6 Byte	2 Byte	46–1500 Byte	4 Byte

B. Delta (Delta time):

This is also an important statistical feature that researchers use to classify traffic that crosses a network. It is called inter-arrival time and calculates the time between the arrival of two successive packets. It is also the time when the previous packet arrived or was captured. In addition, it is also used to measure a network roundtrip, server response time and other delays. The following Equation 14 shows the delta time calculation.

$$\text{Delta time (Inter} - \text{arrival time)} = \text{Arrival Time for packet \#2} - \text{Arrival Time for packet \#1} \quad (14)$$

C. Cumulative Byte:

This statistical feature shows the amount of data that can be transmitted between the sender and the receiver when a large block of data crosses over the network. It is the scale that measures the total bytes that are transmitted in a time interval from the captured traffic. It is also related to the packet length feature for some other calculations. In addition, to the throughput for the network can also be calculated by using it. The throughput value here can be calculated using the following Equation 15:

$$\text{Throughput} = \frac{\text{cumulative byte value}}{\text{packet captured time}} \quad (15)$$

In addition, the cumulative byte can be calculated using the following Equation 16:

$$\text{Cumulative Byte} = \text{Previous Cumulative Byte} + \text{Current Packet Length} \quad (16)$$

D. Relative Time:

This is a statistical feature that displays the elapsed time between the first packet and the current packet, and is sometimes called cumulative

time. It calculates the captured time from the beginning of the capturing process to the last packet that was stopped. It is also related to the delta time feature to enable the calculation using Equation 17:

$$\text{Relative Time} = \text{Previous Relative Time} + \text{Current delta time} \quad (17)$$

The final data that contained the four statistical features for the five applications were applied to the Non-VOIP and VOIP and Multiclasses classification cases. In the next step, the previous dataset was introduced to the WEKA tool and tested on the four ML techniques that were mentioned earlier. These were: Random Forest, J48, MPL and meta.Adaboost(j48). These were tested twice, the first being (Non-VOIP and VOIP classification).The second one was tested on the (Multiclasses classification). Different percentages were obtained for accuracy rate and the other criteria that were mentioned in the previous Section 3.4.

The classification process was carried out as in the previous process. Finally, we achieved different accuracy levels for each of the ML techniques. These were used to detect and classify network traffic in Non-VOIP and VOIP and Multiclasses classification cases, based on the four statistical features which represented the basic characteristics of the different classifiers. This was the end of the processes for traffic classification.

3.6 Wireshark Sniffing Tool

Network technology is still developing and growing day by day. For this reason it has become very important to monitor, maintain and manage network traffic effectively. Therefore, sniffing tools are produced for use in network monitoring areas by people such as network administrators and software engineers. Sniffing tools are used mainly for troubleshooting and other services according to requirements. Wireshark is one of these sniffing tools, and it is a free and open-source packet analyzer. It is used for to capture packets through a live network but can also be used in a network for many purposes, including analysis, troubleshooting, communications protocol development, software and education (Asrodia & Patel, 2012).

The Wireshark sniffing tool can run on different platforms, including Microsoft Windows, other operating systems and Solaris. To use it, the available interface for a particular network must be chosen, and this could include a wireless network connection, a local area connection or other types of interfaces. These interfaces will appear in an interface list apart from the main graphical user interface for the Wireshark tool and then the user can choose one or all of the interfaces according to their requirements. This tool can also open recent files that were opened previously in the same window. In addition, if the user needs help with any subject in Wireshark,

such help is available from the capture helps. All these options and many others are shown in Figure 3.6.

After choosing an interface, other capture options are available in another special window. Other capture options include: selecting filters that can be used to filter the packets according to special constraints; choosing any option from display options; or determining the time in different formats to stop the capture process. It is also possible to choose from the options of name resolution any layers which captured packets, such as network and transport layers and many other options.

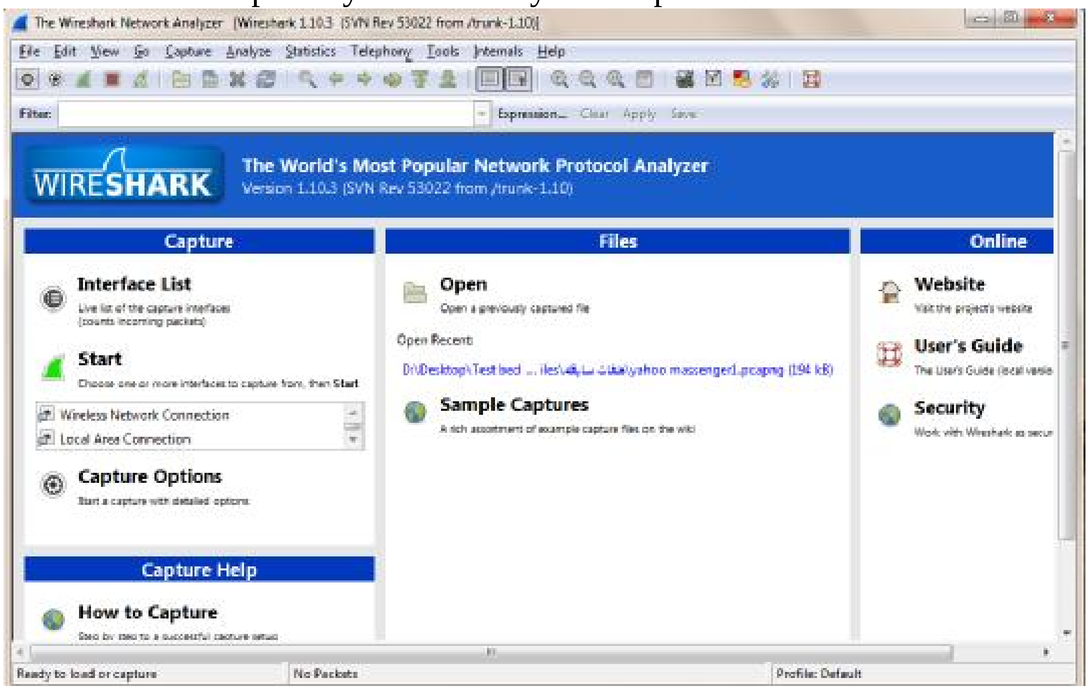


Figure 3.6
Wireshark Sniffing Tool Interface.

The captured file may be viewed after stopping the capture time and the numbers of the captured packets will appear in the window. The colour of the packets will depend on the type of packet according to its protocol and the user can change the colour if they want to. Features that are used for capture are also shown. These features can be selected and changed as the work of the user requires. Figure 3.7 represents the captured file and gives a view of some of the selected features in our work. Other information about capturing packets appears at the bottom of Figure 3.7, such as information about frames, Internet protocols and Ethernet, etc.

To add and remove features, preferences were chosen from the edit list and these were used to define the columns on the left of the window. The features that were required were then selected or other features could have been chosen from the field type in the same window. Figure 3.8 shows the selection of features from the Wireshark sniffing tool.

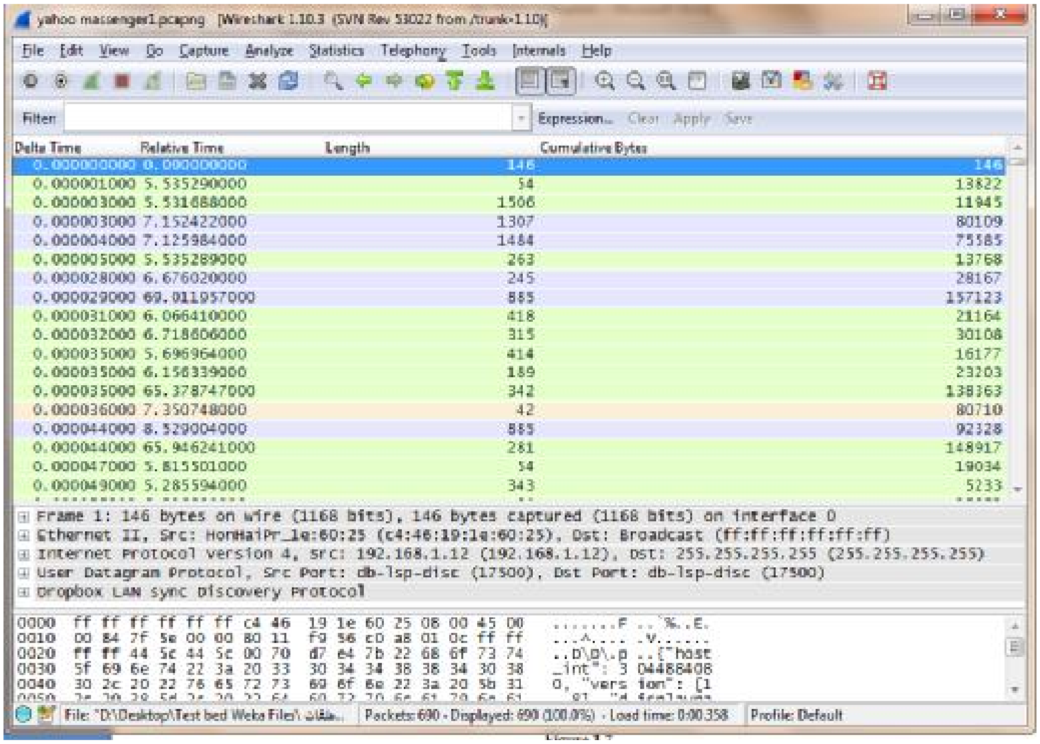


Figure 3.7
Captured File and Selected Features.

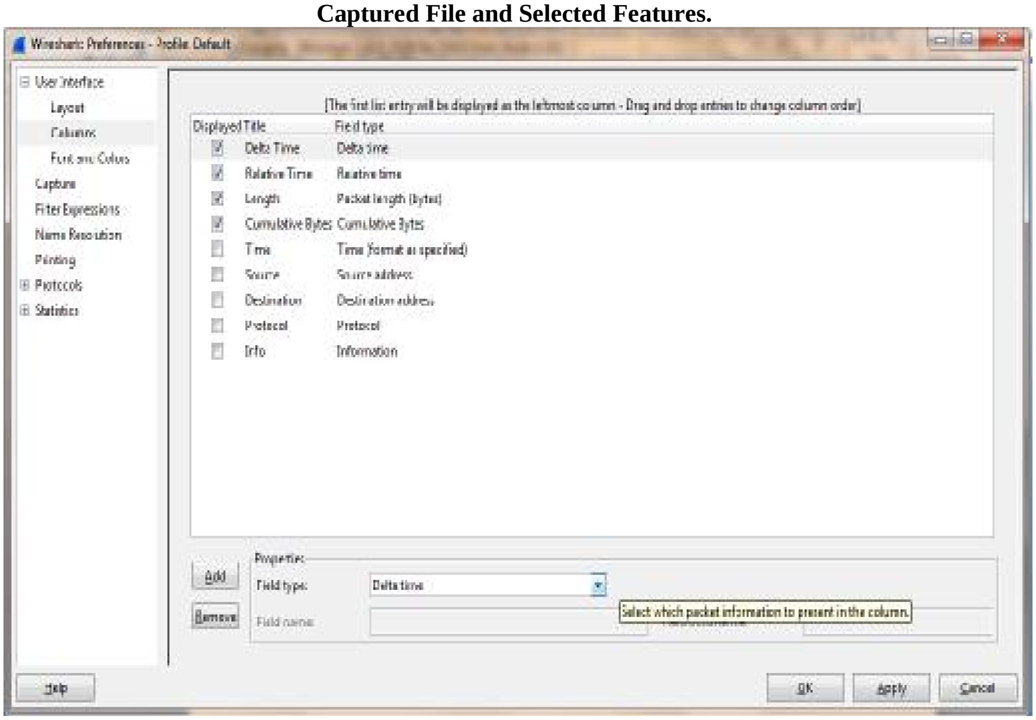


Figure 3.8
Wireshark Features Selection.

CHAPTER FOUR
EXPERIMENTS, DISCUSSION AND RESULTS

4.1 Experiments Set-up

In this thesis, we compiled the generated dataset and applied the experiments on the Windows 7 Ultimate platform, Intel(R) Core(TM)2 Duo CPU T6500 @ 2.10GHz 2.0 GB RAM computers. The Internet speed was 7 Mbps for downloading and 0.91 Mbps for uploading. The software used was Wireshark-win32-1-10-3, TCPView_v3.05 and WEKA version 3.6.12. The TCPView was used to close all background programs in order to capture the correct packets for correct applications only. After that the real time traffic was captured using the Wireshark sniffing tool by applying an experimental testbed.

Real packets were captured from a live network for five applications, namely: PayPal, Gtalk, Yahoo Messenger, Skype and YouTube. These were divided into two cases of classification. The first case was the Non-VOIP and VOIP classes that contained two classes, namely VOIP and Non-VOIP, and each class contained two or three applications. The second case was the Multiclasses classification which included five different classes which means there was a class for each of the previously mentioned applications.

Some the five applications required a secure link, while others needed a reliable connection and the other required speed as a main priority. The dataset was collected from a real experimental testbed as shown in Figure 4.1. This practical experimental network included four PCs, one of which was used to capture the traffic that passed over the network from the five applications using the Wireshark tool. The other PCs 'talk' to each other by means of the VOIP application and the YouTube and PayPal sites were used for different user needs, such as Non-VOIP.

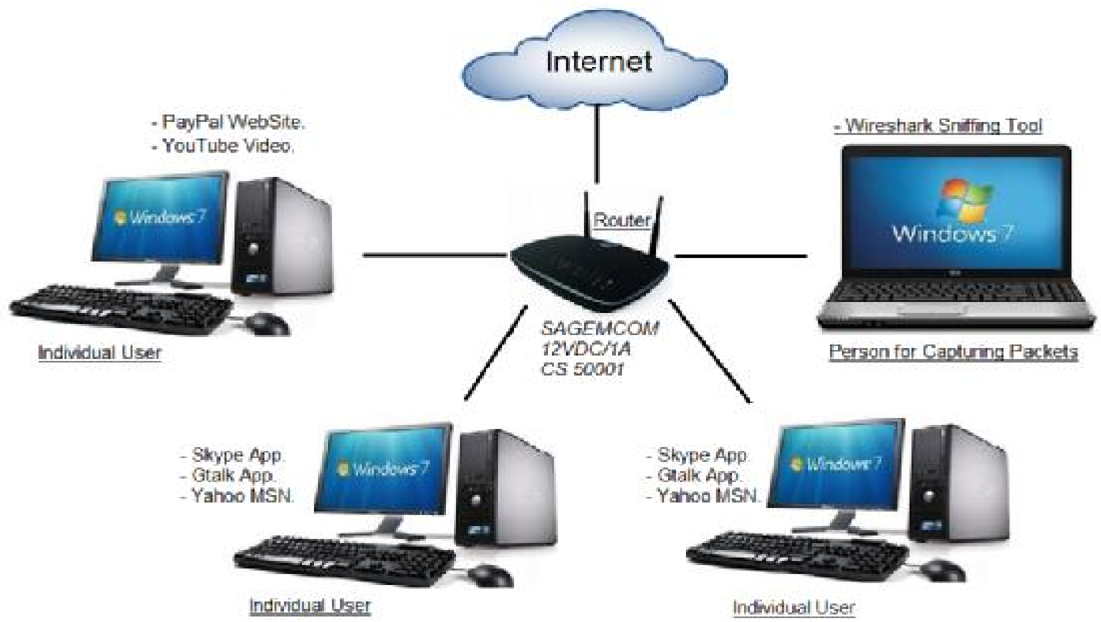


Figure 4.1
Testbed for Real Network.

The classifiers parameters that were used in our experiments are shown in the following tables for each classifier. Parameters values for the meta.Adaboost (j48) classifiers are shown in Table 4.1.

Table 4.1
meta.Adaboost (j48) Parameters Values.

Parameters	**Values**
Number Iteration	10
Seed	1
Weight threshold	100

Parameters values for the Random Forest algorithm are shown in Table 4.2.

Table 4.2
Random Forest Parameters Values.

Parameters	**Values**
Num tree	10
Seed	1
Depth	0 (mean unlimited depth)
Number of execution slots	1

Parameters values for the J48 classifiers are shown in Table 4.3.

Table 4.3
J48 Parameters Values.

Parameters	**Values**
Confidence factor	0.25
Min number object	2
Numbers of Leaves	51
Size of The Tree	101
Num fold	3

Parameters values for the MLP classifiers are shown in Table 4.4.

Table 4.4
MLP Parameters Values.

Parameters	Values
Number of hidden units	500
Random number seed	0
number of threads to use	20

4.2 Non-VOIP and VOIP Results

We tested our dataset to measure the performance of the four ML classifiers. To show the results and compare them, we constructed a confusion matrix and used evaluation criteria as mentioned in Chapter Three. To test the classifiers we used the WEKA toolbox in 10-fold cross validation test mode. The records in our dataset were repeated 10 times for testing and training.

The confusion matrices for classifiers in this work were: meta.Adaboost (j48), RandomForest, J48 and MLP ML classifiers as shown in Table 4.5, Table 4.6, Table 4.7 and Table 4.8 respectively in the case of the Non-VOIP and VOIP classification.

In order to try to achieve better results for the classifiers, the diagonal of the matrices must have higher values than the other upper and lower values of the matrix. In our work, the meta.Adaboost (j48) classifier achieved the highest values, and the MLP classifier the lowest in the case of the Non-VOIP and VOIP. The following matrices display the performance of the classifiers on the different parameters that represent the main components of the confusion matrixes. They are: true positive (TP), false positive (FP), true negative (TN) and false negative (FN).

Table 4.5
Confusion Matrix in (Non-VOIP and VOIP) Classes for meta.Adaboost(j48) algorithm.

	Non-VOIP	VOIP
Non-VOIP	1948	46
VOIP	21	2987

Table 4.6
Confusion Matrix in (Non-VOIP and VOIP) Classes for Random Forest algorithm.

	Non-VOIP	VOIP
Non-VOIP	1924	70
VOIP	33	2975

Table 4.7
Confusion Matrix in (Non-VOIP and VOIP) Classes for J48 algorithm.

	Non-VOIP	VOIP
Non-VOIP	1895	99
VOIP	69	2939

Table 4.8
Confusion Matrix in (Non-VOIP and VOIP) Classes for MLP algorithm.

	Non-VOIP	**VOIP**
Non-VOIP	1728	266
VOIP	57	2951

The results for classified instances of the four classifiers using the WEKA tool are shown in Table 4.9 for Non-VOIP and VOIP classification cases.

Table 4.9
Classified Instances for Non-VOIP and VOIP Classification Case.

Classifiers	**meta.Adaboost(j48)**	**RandomForest**	**J48**	**MLP**
(Correctly Classified Instances)	4935	4899	4834	4679
(Incorrectly Classified Instances)	67	103	168	323
Accuracy (%)	98.6605 %	97.9408 %	96.6413 %	93.5426 %

The average accuracy rate is one of the most important criteria for measuring the performance of the classifiers. Figure 4.2 shows the Average Accuracy rate (AA) in the case of Non-VOIP and VOIP classification. The highest values for the AA indicate a good prediction model for the selected classifiers and the lowest values indicate a bad prediction model for the selected classifiers. The results for AA are discussed below in the case of Non-VOIP and VOIP classification. We also calculated the Average Accuracy (AA) using Equation 10.

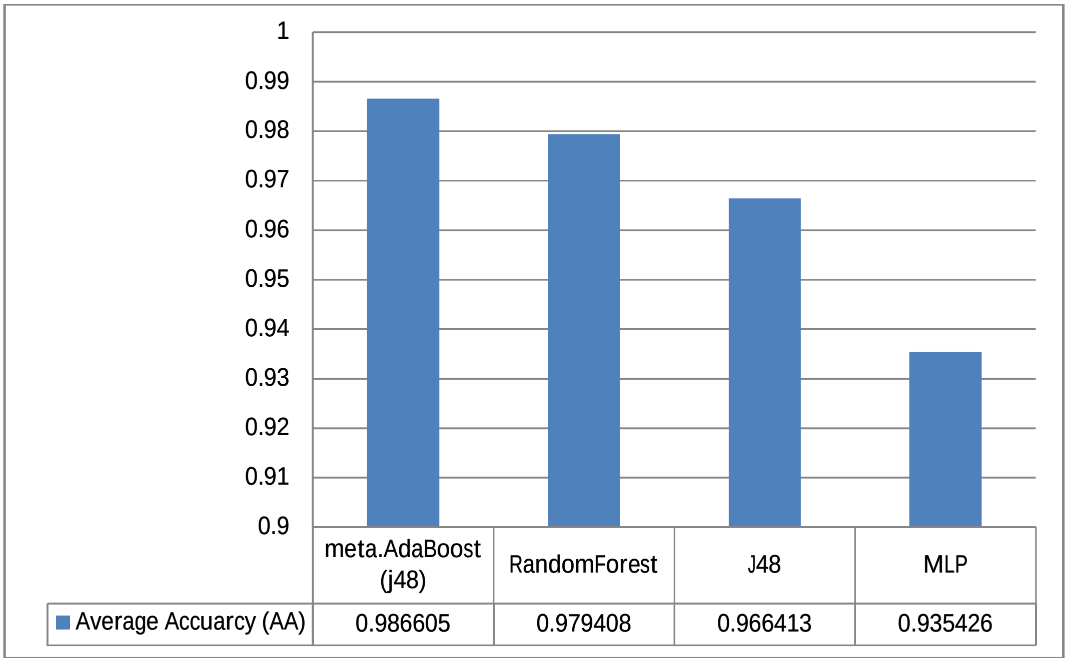


Figure 4.2
Average Accuracy Rate for Non-VOIP and VOIP Classes.

The meta.Adaboost (j48) classifier achieved the highest Average Accuracy (AA) rate in the case of Non-VOIP and VOIP classification: 98.6605%, which represents the best result. The MLP classifier achieved the lowest Average Accuracy (AA) rate in the case of Non-VOIP and VOIP: 93.5426%. The results for other two classifiers were close to each other in the case of Non-VOIP and VOIP classification. Random Forest and J48 achieved results equal to 97.9408% and 93.5426% respectively.

In the classification process we need indicators and criteria to measure the performance of the proposed ML classifiers. The following comments are presented pertaining to the results for this section and the next section:

1- True Positive (TP) for correctly classified, e.g. classifying the VOIP as VOIP.
2- True Negative (TN) is an indicator that refers to what has been correctly classified, like TP.
3- False Negative (FN) for incorrectly classified, e.g. classifying VOIP as Non-VOIP.
4- False Positive (FP) is an indicator that refers to what has been incorrectly classified, like FN.

The following Figure 4.3 shows the Recall Average in the case of Non-VOIP classification. This represents the percentage of the correct classification instances from the actually correct classification instances. In the case of Non-VOIP and VOIP, the meta.Adaboost(j48) classifier achieved the highest rate of 0.987%. The MLP classifier achieved the lowest recall rate of 0.935% for Non-VOIP and VOIP classification. There

was no great difference between them the other two classifiers in the case of Non-VOIP and VOIP classification .The results for Random Forest and J48 were equal to 0.979% and 0.966% respectively.

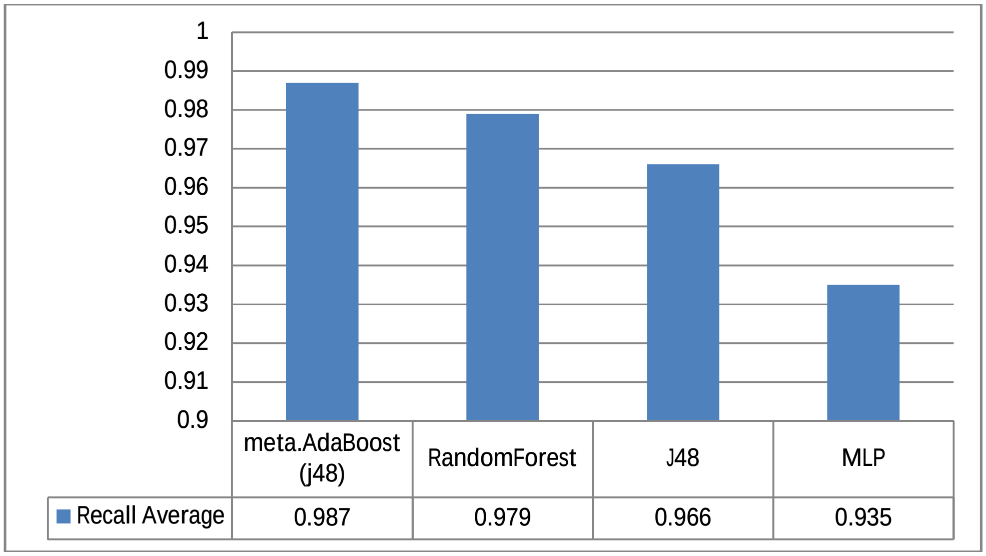


Figure 4.3
Recall Average for Non-VOIP and VOIP Classes.

FPR is another performance indicator as mentioned earlier in this section. Figure 4.4 shows the FPR for Non-VOIP and VOIP classification.

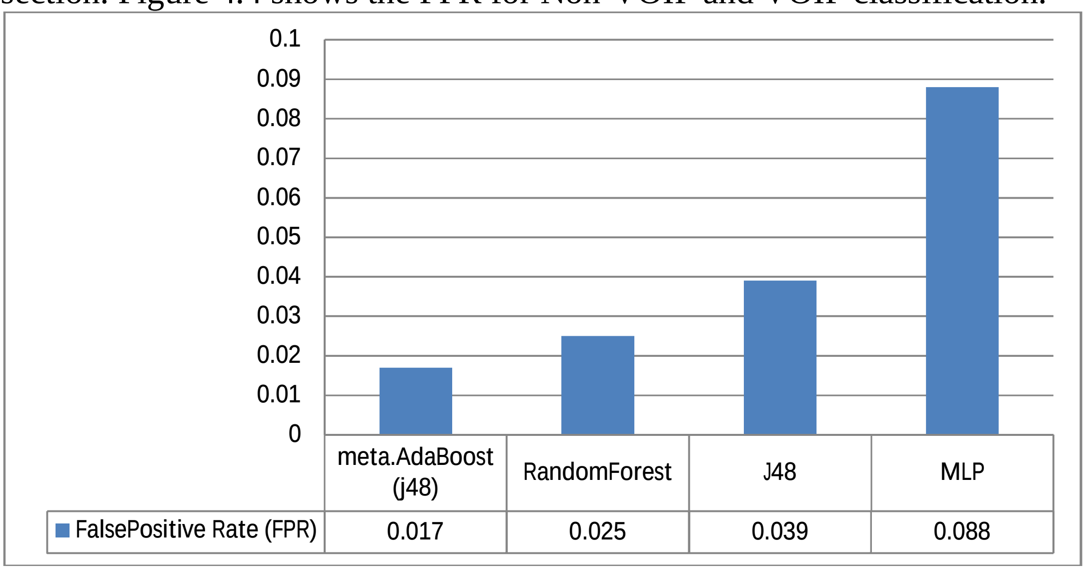


Figure 4.4
False Positive Rate for Non-VOIP and VOIP Classes.

The lowest rate in FPR for classifiers means that, in this instance, the classifier has been classified incorrectly. Therefore, the lowest values are good values, like the first classifier, meta.Adaboost(j48), which achieved the lowest values for Non-VOIP and VOIP classification, equal to 0.017%. MLP achieved the highest FPR for Non-VOIP and VOIP classification,

equal to 0.088%. That means that this classifier was classified incorrectly in a large number of instances (e.g. classified Non-VOIP as VOIP). The values for the other two classifiers, Random Forest and J48, were close to each other and equal to 0.025% and 0.039% respectively. We also calculated the FPR using Equation 7.

In this work, we tested and classified 5002 records in our generated dataset. Through this process the classifiers were classified in the previous instances correctly and incorrectly. The following Figure 4.5 displays the number of incorrectly classified instances for Non-VOIP and VOIP classification.

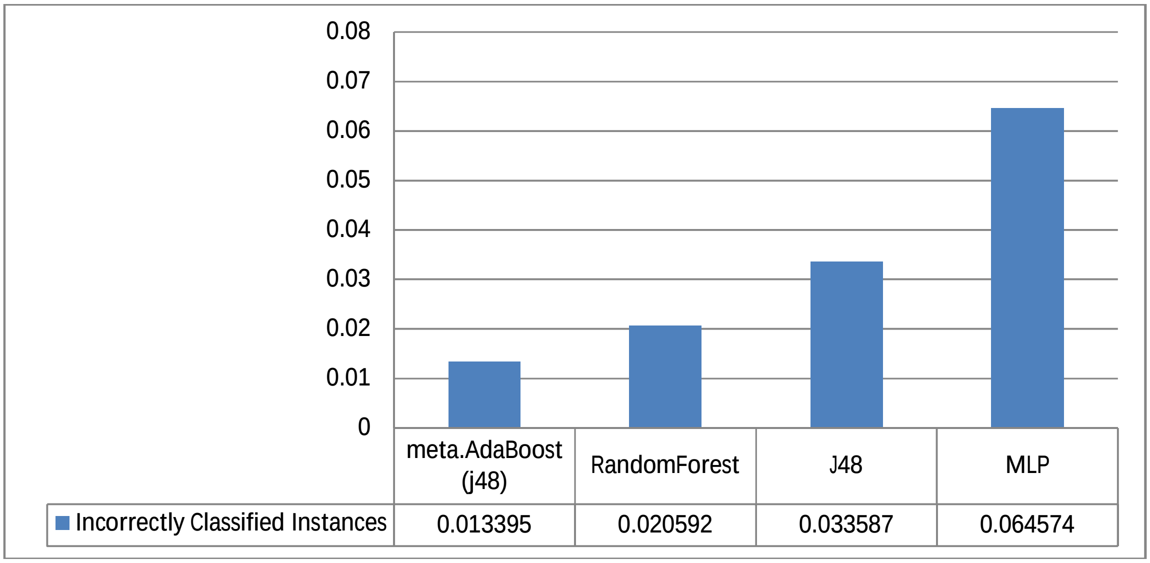


Figure 4.5
Incorrectly Classified Instances for Non-VOIP and VOIP Classes.

The meta.Adaboost(j48) achieved the lowest number of incorrect instances for Non-VOIP and VOIP classification, equal to 0.013395%. The low number of incorrect instances means a high level of accuracy for the classifier used. The highest number of incorrect instances achieved by the MLP classifier was equal to 0.064574%. The high number of incorrect instances means a low level of accuracy for the classifier. We can also see that there was no big difference between the other two classifiers, RandomForest and J48.

Precision is one of the most important indicators in measuring the performance of classifiers. It means that the percentage of records that were classified as correct are actually correct (e.g. classifying VOIP packets as VOIP). This can be calculated using Equation 12. Figure 4.6, Figure 4.7, Figure 4.8 and Figure 4.9 illustrate the precision results for meta.Adaboost(j48), RandomForest, J48 and MLP Classifier respectively in Non-VOIP and VOIP classification.

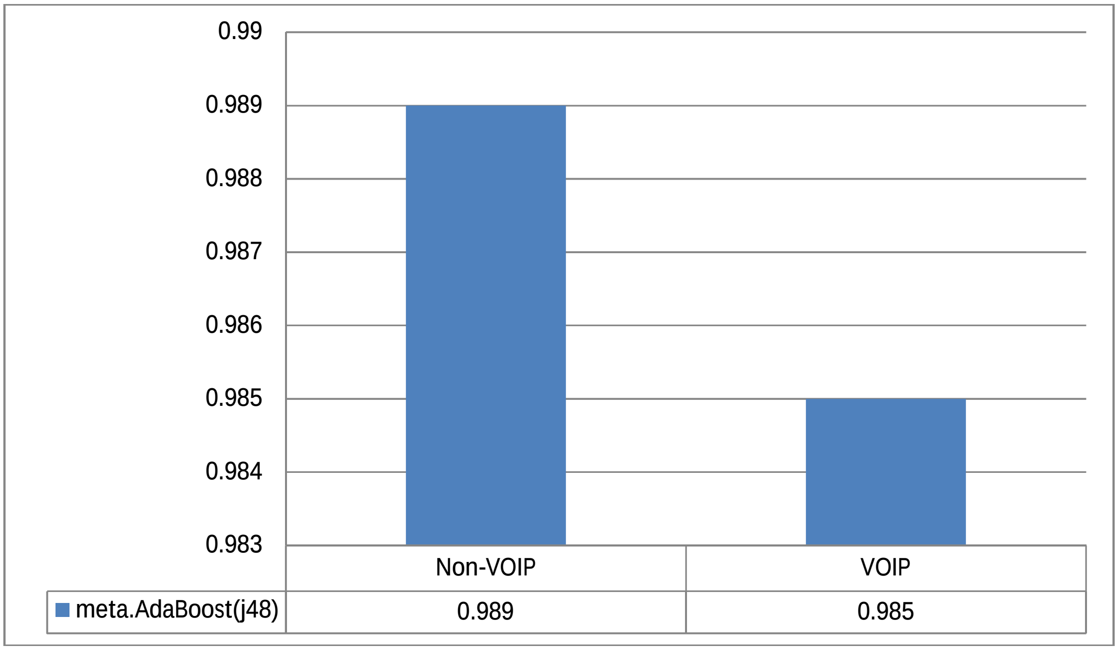


Figure 4.6
Precision of meta.Adaboost(j48) for Non-VOIP and VOIP Classes.

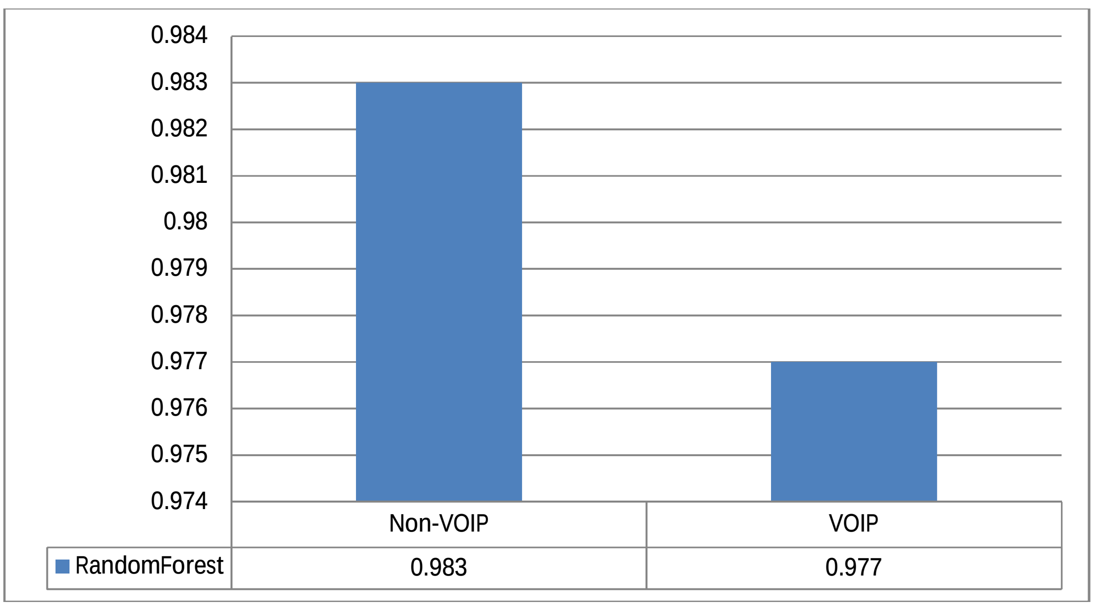


Figure 4.7
Precision of RandomForest for Non-VOIP and VOIP Classes.

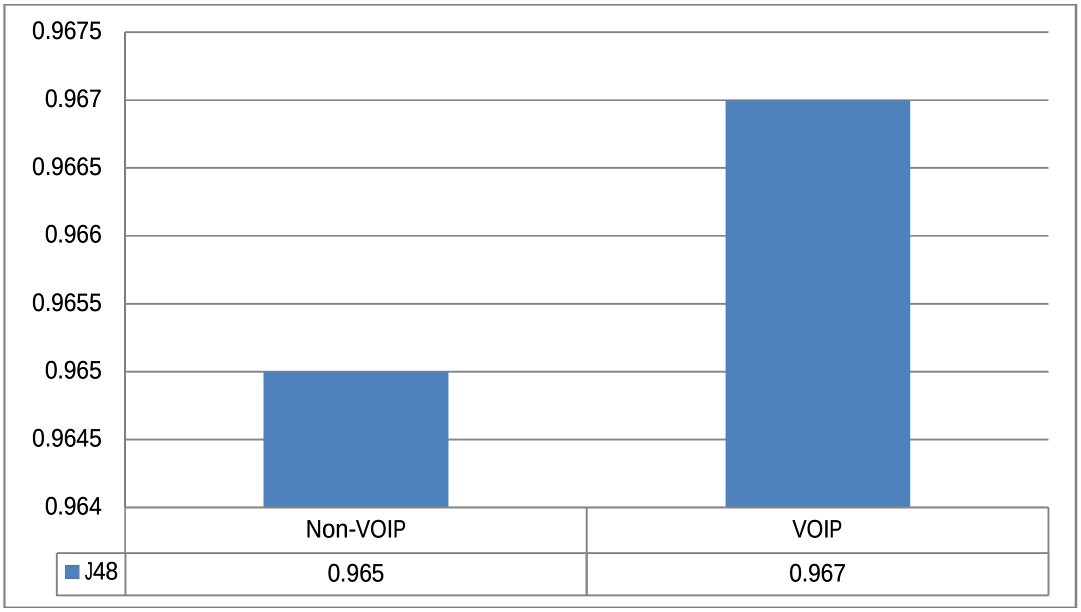


Figure 4.8
Precision of J48 for Non-VOIP and VOIP Classes.

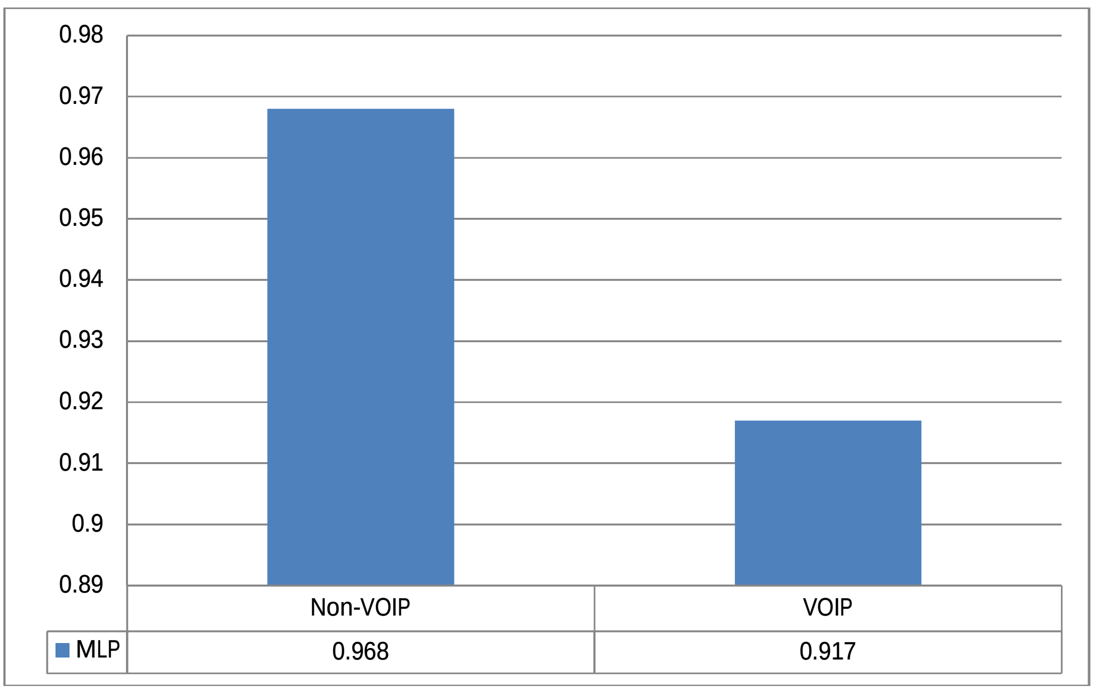


Figure 4.9
Precision for MLP for Non-VOIP and VOIP Classes.

In the precision results for the case of Non-VOIP and VOIP classification we can see the difference between the two classes for each classifier. In meta.Adaboost(j48), Random Forest and MLP classifiers we can see that the Non-VOIP class achieved the highest precision rate than VOIP class refers to correctly classified instances. However, in the J48

classifiers we can see also that the VOIP class achieved the highest precision rate than Non-VOIP class refers to correctly classified instances. We can calculate precision by using Equation 12.

The F-Measure is a measure used to test accuracy rate for the performance, combining both recall and precision measures. To calculate the F-Measure we used Equation 13. The following Figure 4.10 shows F-Measure values for Non-VOIP and VOIP classification case.

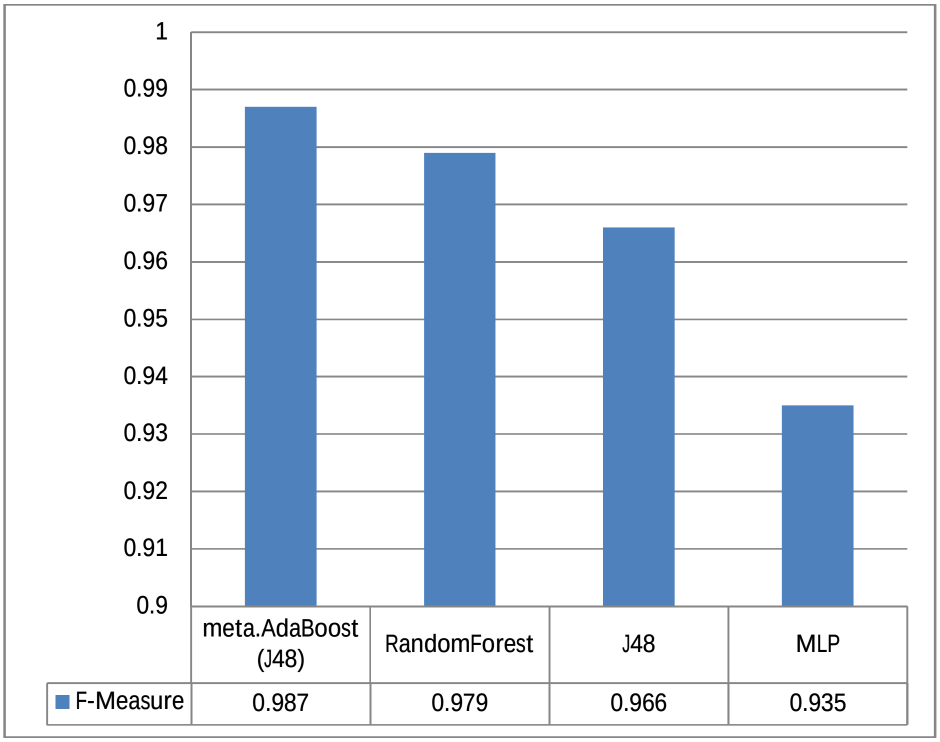


Figure 4.10
F-Measure for Non-VOIP and VOIP Classes.

The meta.Adaboost(j48) achieved the highest rate in the F-Measure for Non-VOIP and VOIP classification: 0.987%,which means that it was classified as the best, while the MLP classifier achieved the lowest rate in the F-Measure for Non-VOIP and VOIP classification: 0.935%,which means that it was classified as low. For Non-VOIP and VOIP classification, the Random Forest and J48 classifiers achieved convergent percentages. Their results equaled 0.979% and 0.966% respectively.

Figure 4.11 shows the RMSE for the selected classifier algorithms for Non-VOIP and VOIP classification. Root mean squared errors are frequently considered as a method used to measure the differences between predicted values and real values for the classification process. If low values are achieved, that means that the evaluation process is mostly accurate, and also that the error rate was reduced according to their values (e.g. zero value means no error).

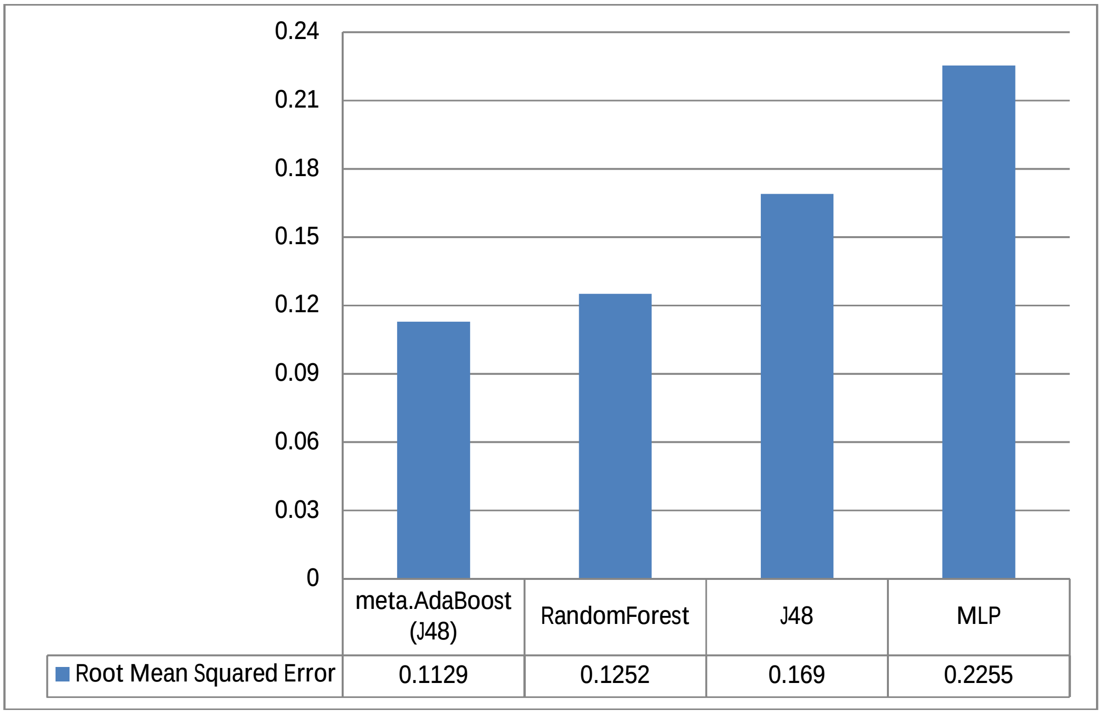


Figure 4.11
Root Mean Squared Errors for Non-VOIP and VOIP Classes.

As shown in the previous figure, the meta.Adaboost(j48) classifier achieved the lowest values for Non-VOIP and VOIP classification: An error rate of 0.1129 error. That indicates a reduced number of errors using meta.Adaboost(j48) for Non-VOIP and VOIP classification. MLP classifiers achieved the highest values for Non-VOIP and VOIP classification: an error rate of 0.2255. That also means that this classifier achieved the highest number of errors among all the other classifiers. We can also see that the Random Forest classifier achieved a percentage convergent with the lowest error rate. As we can see in the previous figure, the error rate for J48 is closer to the rates for Random Forest.

The area under the ROC (Receiver Operating Characteristic) curves for meta.Adaboost (j48), RandomForest, J48 and MLP are shown in Figure 4.12, Figure 4.13, Figure 4.14 and Figure 4.15 for Non-VOIP and VOIP classification respectively. These show the relationship between True Positive (TP) and False Positive (FP) as mentioned in detail in Chapter Three. The following areas under the ROC curve figures are to measure accuracy in particular. To know and understand the area under the ROC curves, special groups to evaluate performance are as follows:

a) 1 - 0.900 = excellent.
b) 0.890 - 0.800 = good.
c) 0.790 – 0.700 = fair.
d) 0.690 – 0.600 = poor.
e) 0.590 – 0.500 = fail.

The area under the ROC curves achieved the highest values for True Positive (TP) and False Positive (FP), and so the accuracy is excellent, like the first group. The following figures also show the highest values for the data in our dataset. In addition, we made comparisons between all the developed ML classifiers.

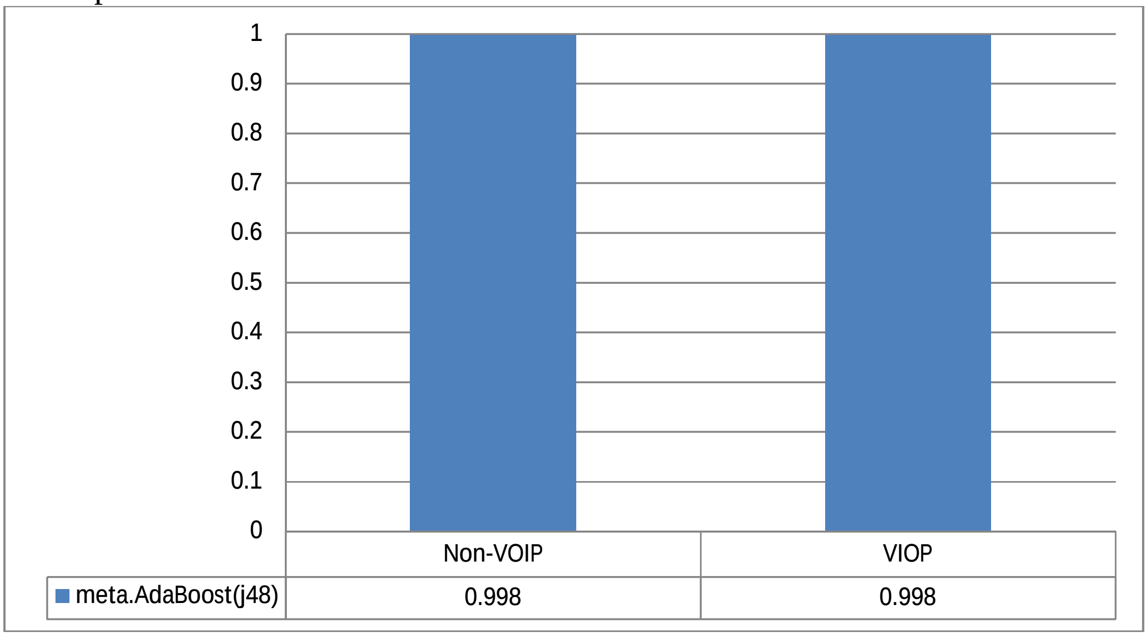


Figure 4.12
Area under ROC for Non-VOIP and VOIP Classes of meta.Adaboost(j48).

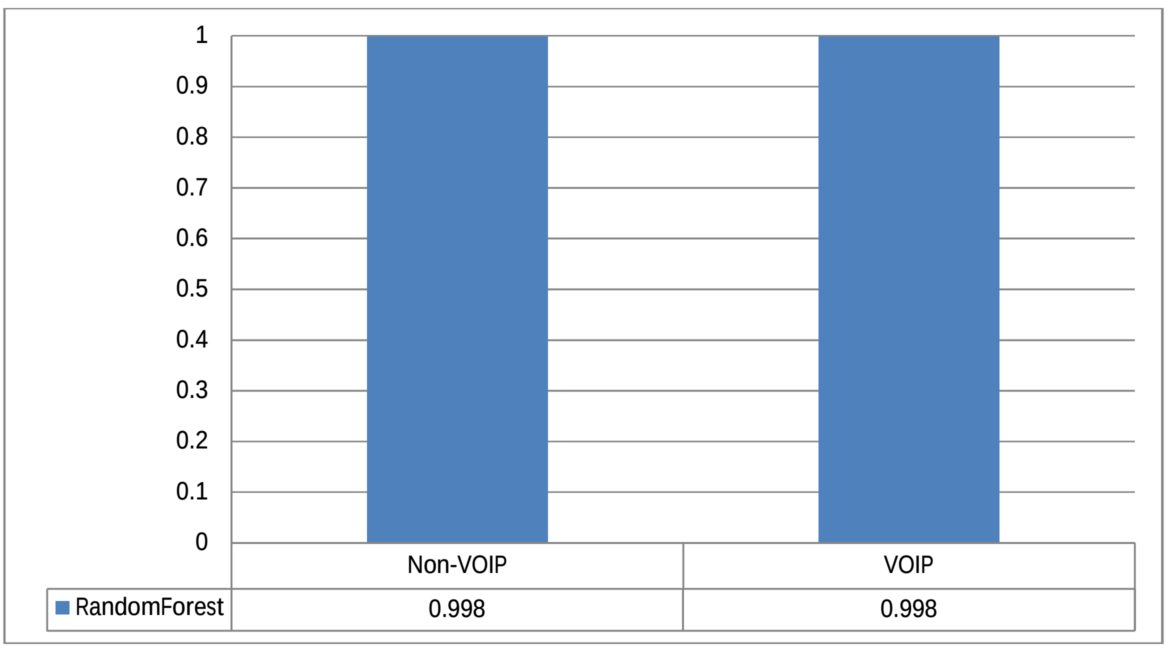


Figure 4.13
Area under ROC for Non-VOIP and VOIP Classes of Random Forest.

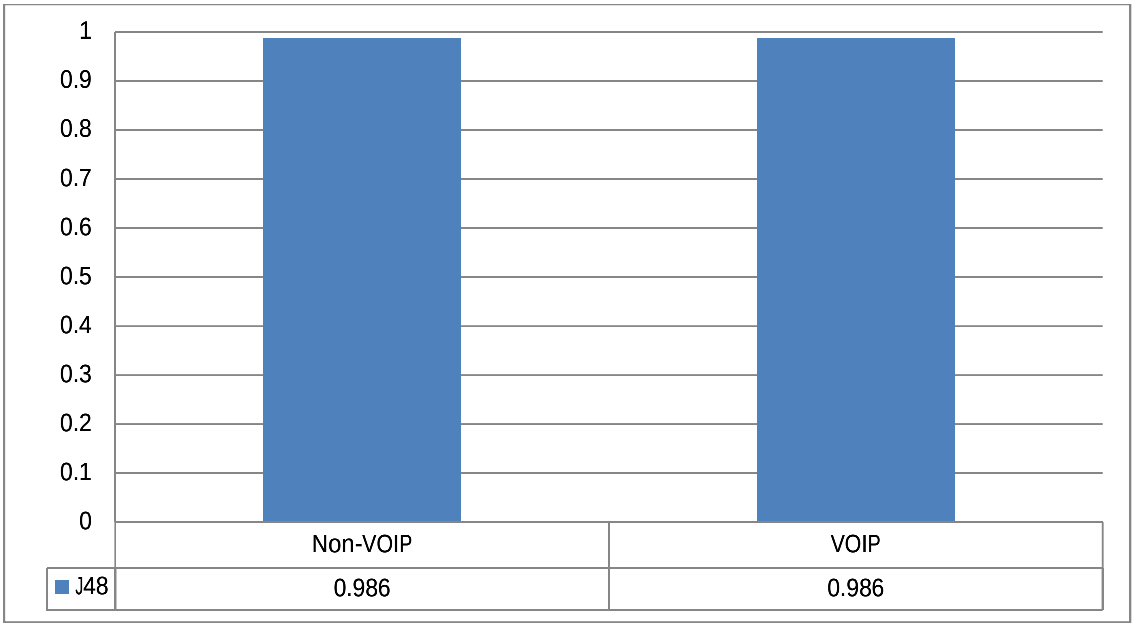


Figure 4.14
Area under ROC for Non-VOIP and VOIP Classes of J48.

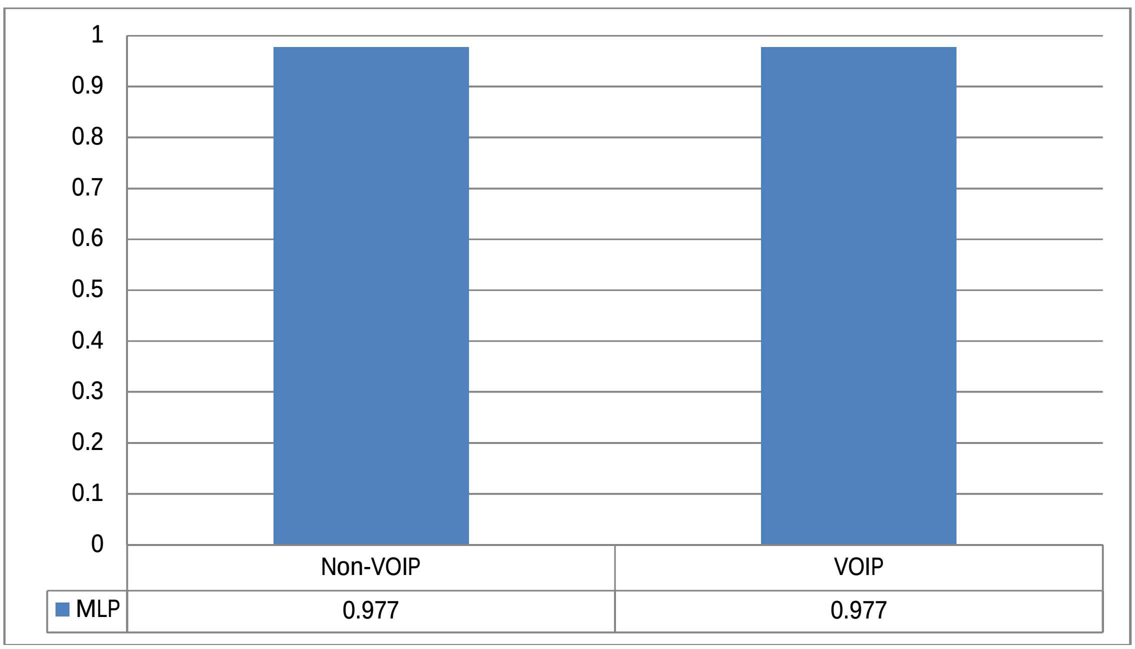


Figure 4.15
Area under ROC for Non-VOIP and VOIP Classes of MLP.

The ROC curve values are equal for classes within each ML classifier. Random Forest and meta.Adaboost(j48) achieved higher ROC curve values in Non-VOIP and VOIP classification for each class. They achieved 0.998% for all classes and that means excellent prediction for classes in the classification process. On other hand, MLP achieved the lowest ROC curve values in Non-VOIP and VOIP classification. Each of the two classes was equal to 0.977%, which means poor prediction for classes in the classification process. J48 classifier achieved the middle

value between the highest and the lowest values that was equal to 0.986% for the two classes.

Each classifier needs time for the building of a training model using the WEKA toolbox. Figure 4.16illustrates the time that the selected classifier takes to build a training model for Non-VOIP and VOIP classification.

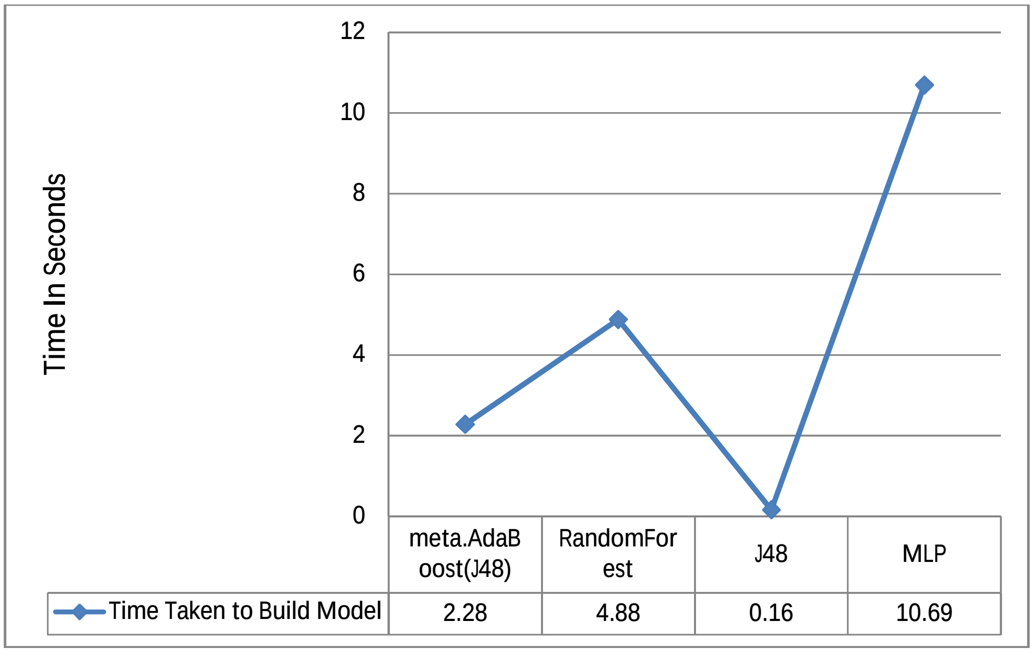


Figure 4.16
Time Taken to Build a Model for Non-VOIP and VOIP Classes.

The J48 classifier achieved the shortest time to build a model for our dataset that contained 5002 records in Non-VOIP and VOIP classification: 0.16 seconds, which is the fastest time as compared to the other classifiers. On the other hand, MLP took the longest time to build a model for our dataset for Non-VOIP and VOIP classification: 10.69 seconds, which is the slowest time as compared to the other classifiers.

For other classifiers, meta.Adaboost(j48) achieved the second shortest time after the J48 classifier for Non-VOIP and VOIP classification: 2.28 seconds. Random Forest achieved the third shortest time after meta.Adaboost(j48) and J48's time to build a model came before the MLP classifier: 4.88 seconds.

4.3 Multiclasses Results

As mentioned earlier, confusion matrices were used to measure the performance of the ML classifiers that we used here. The following Table 4.10, Table 4.11, Table 4.12 and Table 4.13 show the confusion matrices for Multiclasses classification respectively.

Table 4.10
Confusion Matrix in Multiclasses for meta.Adaboost(j48) algorithm.

	PayPal	Gtalk	Yahoo Messenger	Skype	YouTube
PayPal	959	5	0	34	2
Gtalk	6	983	3	8	0
Yahoo Messenger	0	3	995	1	0
Skype	14	6	1	988	0
YouTube	0	0	0	2	992

Table 4.11
Confusion Matrix in Multiclasses for Random Forest algorithm.

	PayPal	Gtalk	Yahoo Messenger	Skype	YouTube
PayPal	954	10	0	34	2
Gtalk	7	985	0	7	1
Yahoo Messenger	0	5	993	1	0
Skype	20	10	1	978	0
YouTube	3	0	0	0	991

Table 4.12
Confusion Matrix in Multiclasses for J48 algorithm.

	PayPal	Gtalk	Yahoo Messenger	Skype	YouTube
PayPal	932	17	0	48	3
Gtalk	18	960	7	14	1
Yahoo Messenger	2	5	992	0	0
Skype	32	18	0	959	0
YouTube	2	0	0	0	992

Table 4.13
Confusion Matrix in Multiclasses for MLP algorithm.

	PayPal	Gtalk	Yahoo Messenger	Skype	YouTube
PayPal	655	52	0	290	3
Gtalk	6	818	64	112	0
Yahoo Messenger	0	29	967	3	0
Skype	65	156	0	783	5
YouTube	1	3	0	3	987

The results for classified instances of the four classifiers using the WEKA tool are shown in Table 4.14 for Multiclasses classification cases.

Table 4.14
Classified Instances for Multiclasses Classification Case.

Classifiers	meta.Adaboost(j48)	RandomForest	J48	MLP
(Correctly Classified Instances)	4917	4901	4835	4210
(Incorrectly Classified Instances)	85	101	167	792
Accuracy (%)	98.3007%	97.9808 %	96.6613 %	84.1663 %

The average accuracy rate is one of the most important criteria used to measure the performance of classifiers. Figure 4.17 shows the Average Accuracy rate (AA) for Multiclasses classification. The highest values of AA indicate a good prediction model for the selected classifiers and the lowest values indicate a bad prediction model for the selected classifiers. The results for AA are discussed below for Multiclasses classification. We also calculated Average Accuracy (AA) using Equation 10, as mentioned earlier.

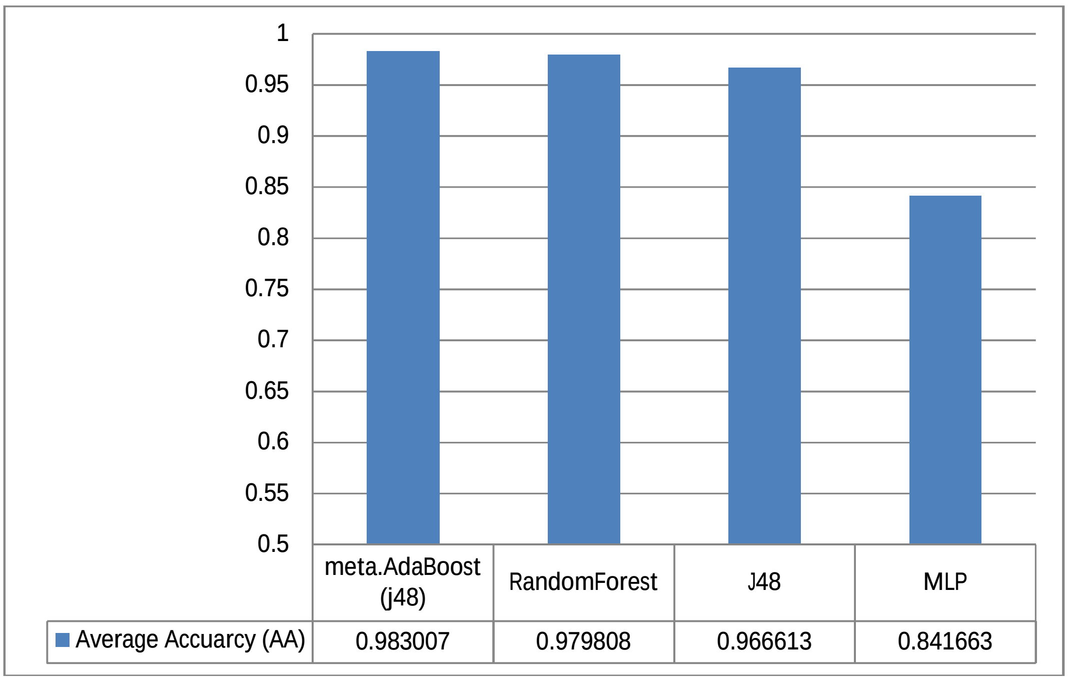


Figure 4.17
Average Accuracy Rate for Multiclasses.

The meta.Adaboost (j48) classifier achieved the highest Average Accuracy (AA) rate for Multiclasses classification: 98.3007%. The MLP classifier achieved the lowest Average Accuracy (AA) rate for Multiclasses classification: 84.1663%. The results for the other two classifiers were

close to each other – Random Forest and J48 achieved results equal to 97.9808% and 96.6613% respectively.

The following Figure 4.18 shows the Recall Average for Multiclasses classification, which represents the percentage of correct classification instances from them were actually correct. In Multiclasses classification, the meta.Adaboost(j48) classifier achieved the highest recall rate: 0.983%.The MLP classifier achieved the lowest recall rate for Multiclasses classification: 0.935%. There was no great difference between the other two classifiers in Multiclasses classification, with the recall rate for Random Forest and J48 equal to 0.979% and 0.966% respectively.

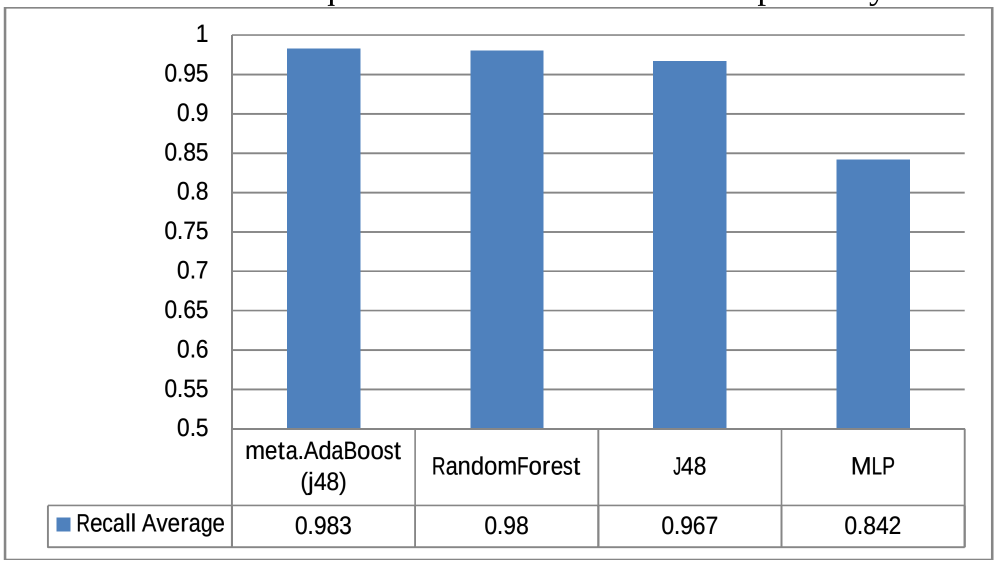


Figure 4.18
Recall Average for Multiclasses.

As mentioned earlier, FPR is another performance indicator to measure the performance of ML classifiers. Figure 4.19 shows the FPR for Multiclasses classification.

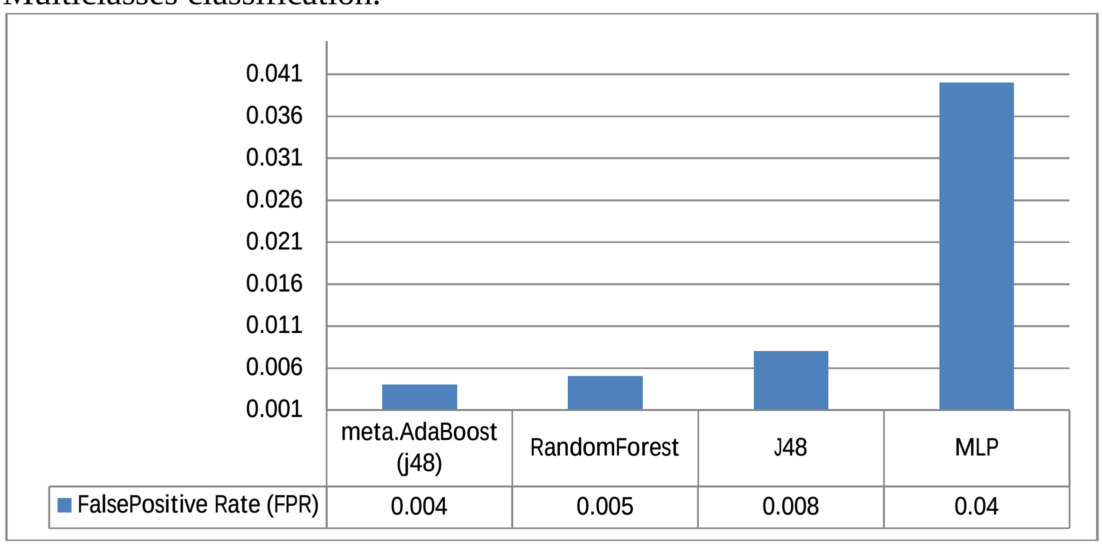


Figure 4.19
False Positive Rate for Multiclasses.

We used FPR to measure the performance of ML classifiers. As mentioned earlier, the lowest rate of FPR for classifiers means that there were some instances are incorrect classification. Therefore, the lowest values are good values like the first classifier, meta.Adaboost(j48), which achieved the lowest values for Multiclasses classification: 0.004%. MLP achieved the highest FPR for Multiclasses classification: 0.04%. That means that there were incorrect classifications in a large number of instances (e.g. classifying the Skype application as Yahoo Messenger). For the other two classifiers, the Random Forest value was close to that of meta.Adaboost(j48), with very little difference, equal to 0.005% and the J48 classifier achieved a value equal to 0.008%, very close to the Random Forest classifier. We calculated FPR using Equation 7.

In the following Figure 4.20 the number of incorrect classified instances for Multiclasses classification is shown. By means of this process the classifiers classified the dataset instances correctly and incorrectly, as mentioned previously.

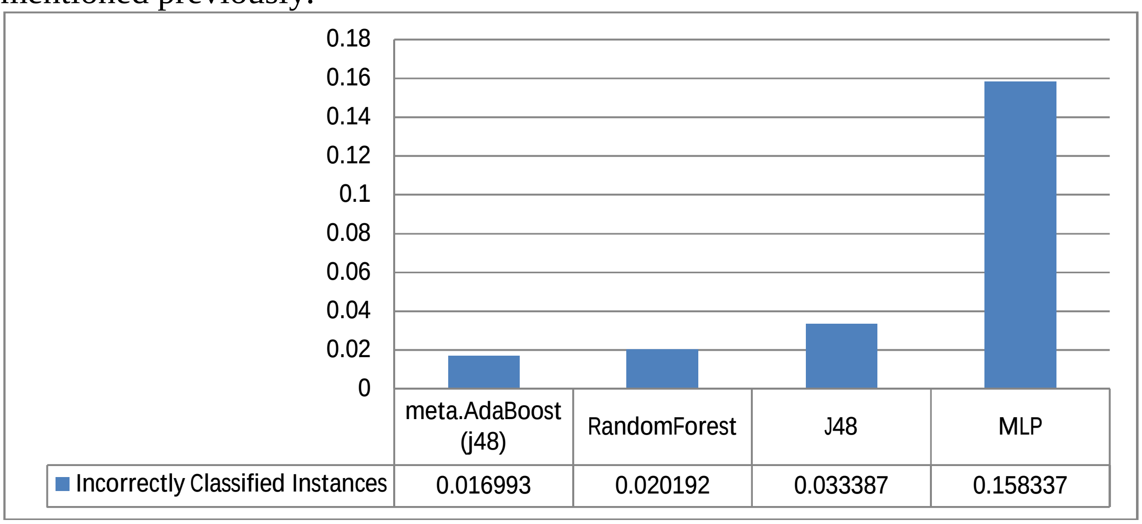


Figure 4.20
Incorrectly Classified Instances for Multiclasses.

The meta.Adaboost(j48) classifier achieved the lowest number of incorrect instances for Multiclasses classification: 0.016993%. A low number of incorrect instances mean better accuracy for this classifier. The highest number of incorrect instances was achieved by the MLP classifier: 0.158337%. A high number of incorrect instances mean bad accuracy for the classifier. We can also see that there was no big difference between the other two classifiers, Random Forest and J48, within this classification.

The following Figure 4.21, Figure 4.22, Figure 4.23 and Figure 4.24 show the precision results for Multiclasses classification.

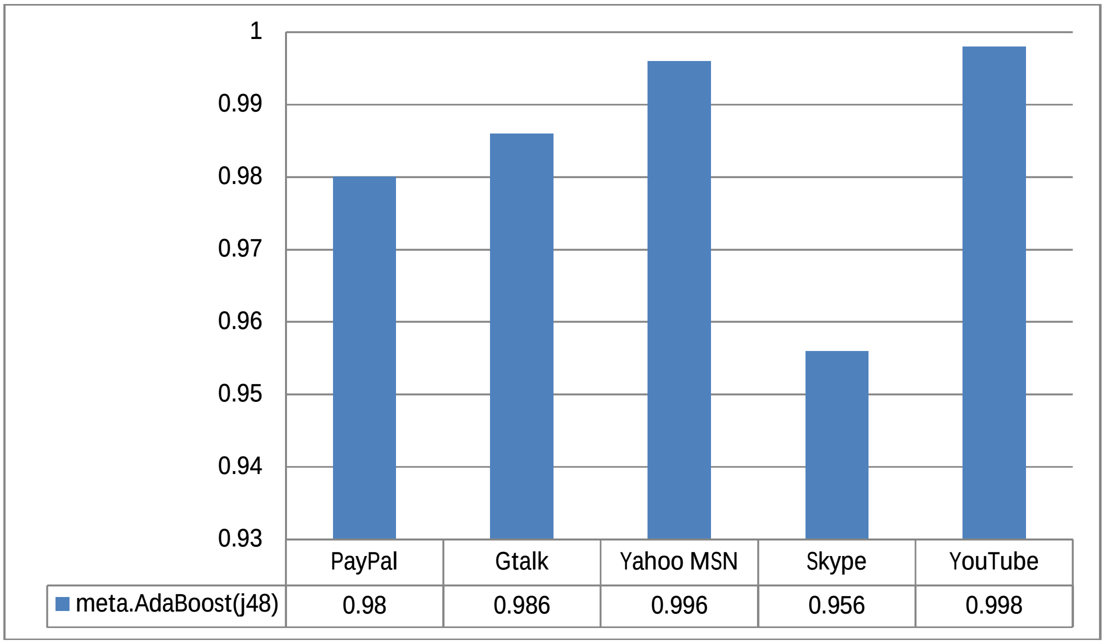


Figure 4.21

Precision for meta.Adaboost(j48) **for Multiclasses.**

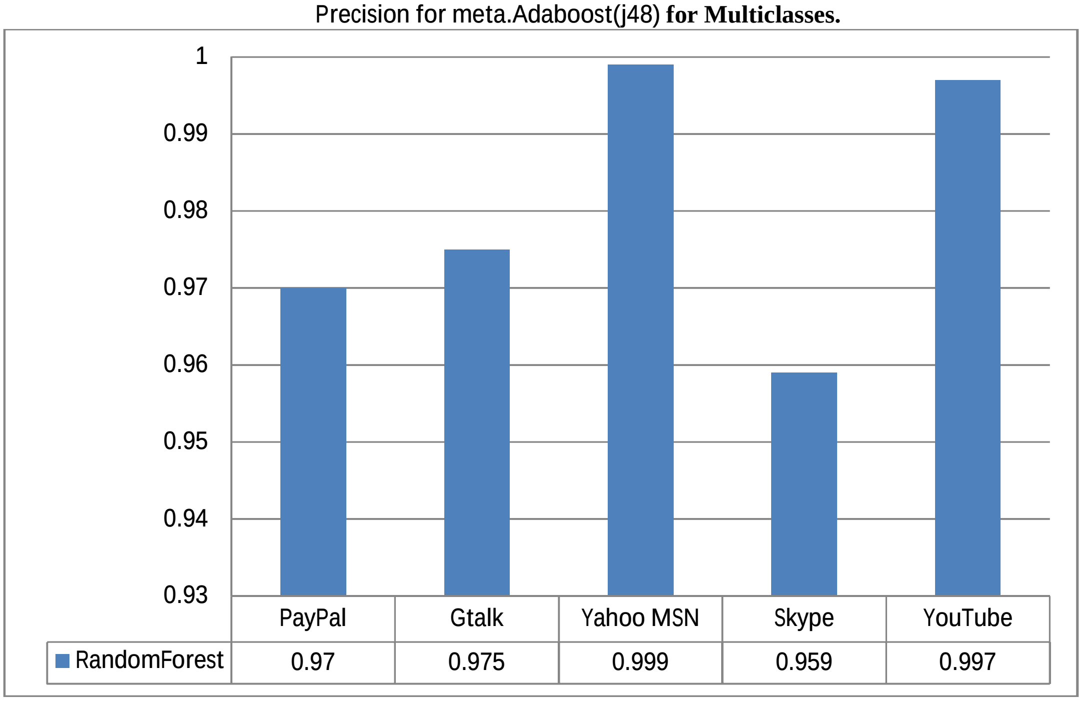


Figure 4.22

Precision for Random Forest **for Multiclasses.**

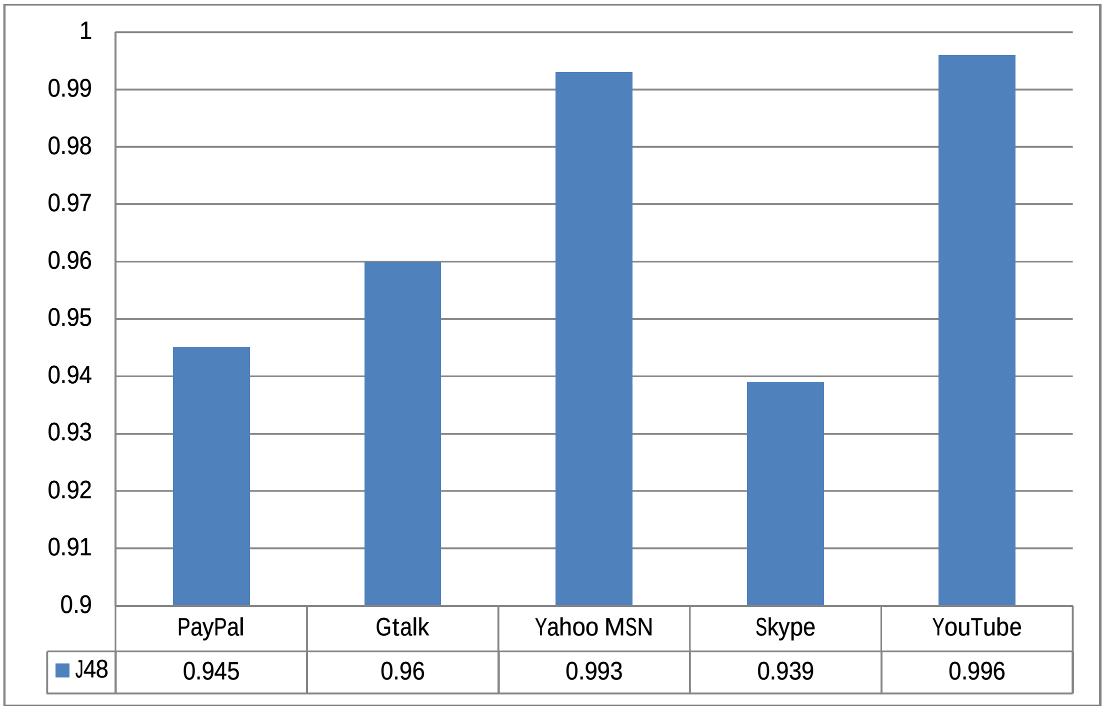


Figure 4.23
Precision for J48 **for Multiclasses.**

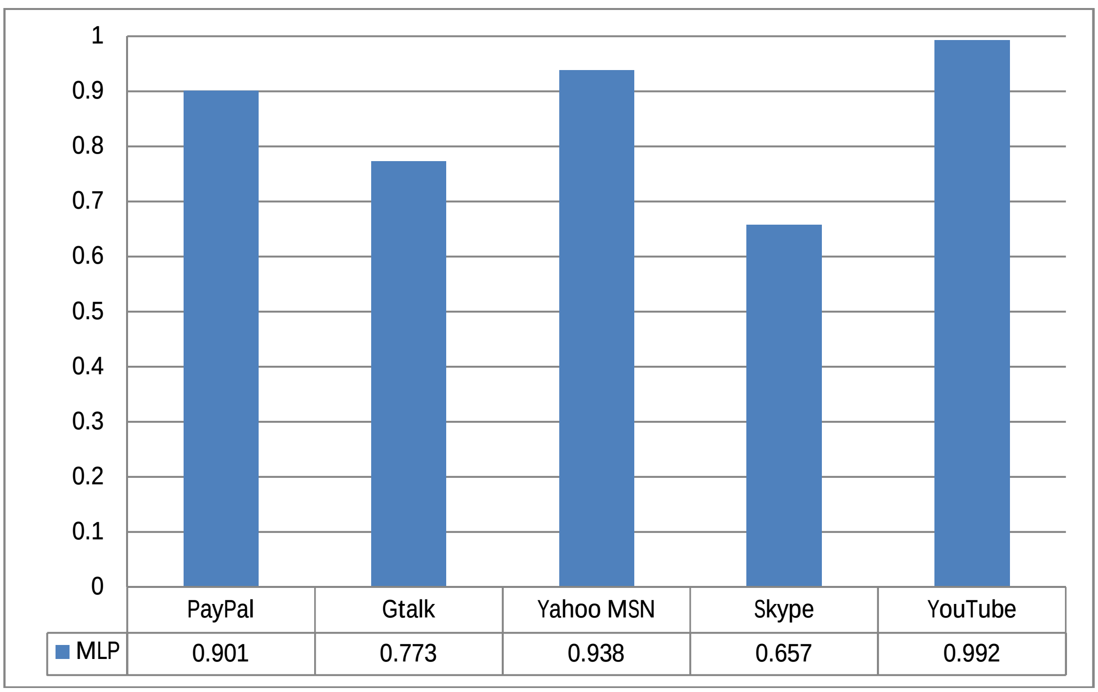
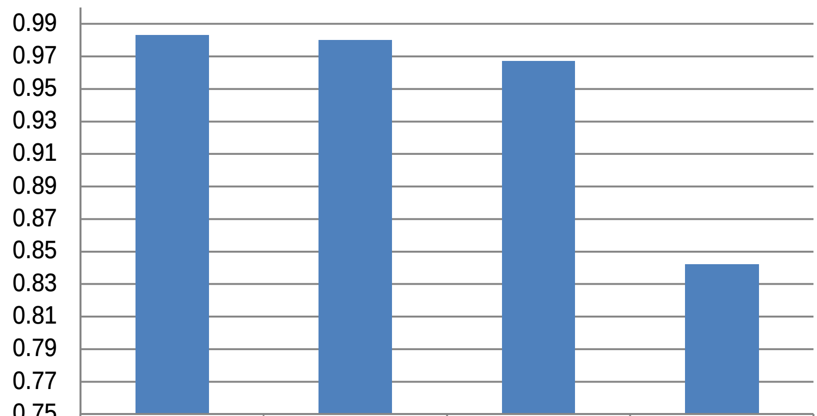


Figure 4.24
Precision for MLP **for Multiclasses.**

Precision results are different for each classifier in their classes. The YouTube class achieved the highest precision results of all the classifiers except for Random Forest, with a simple difference as compared with the Yahoo Messenger class, and the same thing between all classes in each

classifier. The Skype class achieved the lowest precision results of all the classifiers independently and between all classes in each classifier. The Yahoo Messenger class achieved second place for the highest precision results in simple different after the YouTube class except for Random Forest, as mentioned above. Precision values for the PayPal and Gtalk classes were very close to each other, as shown in the previous figures.

The F-measure is a combination of recall and precision rates measures. It is used to measure the accuracy of the ML classifier's performance, as mentioned previously. To calculate the F-Measure we used Equation 13. The following Figure 4.25 shows F-Measure values for Multiclasses classification.



	meta.AdaBoost (J48)	RandomForest	J48	MLP
F-Measure	0.983	0.98	0.967	0.842

Figure 4.25
F-Measure for Multiclasses.

The meta.Adaboost(j48) achieved the highest rates in the F-Measure for Multiclasses classification: 0.983 %.The MLP classifier achieved the lowest rate in the F-Measure for Multiclasses classification: 0.842%,while Random Forest and J48 classifiers achieved percentages convergent to each other. Their results were equal to 0.98% and 0.967% respectively, the result for Random Forest classifier being closer to meta.Adaboost(j48) than J48 classifier.

Figure 4.26 shows RMSE for the classifiers in Multiclasses classification. This was used to measure the differences between predicted values and real values in the classification process. Low values mean the evaluation process is mostly accurate, and that errors are reduced according to their values (e.g. zero value means no errors) as mentioned earlier.

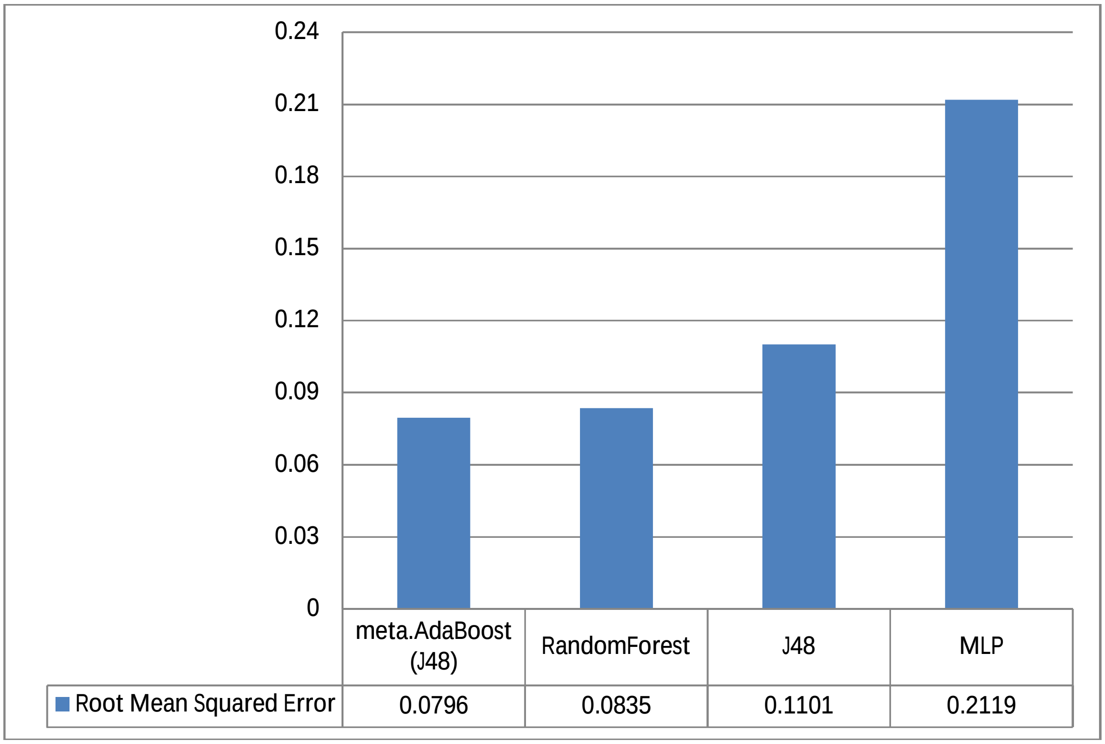


Figure 4.26
Root Mean Squared Errors for Multiclasses.

The previous figure shows that meta.Adaboost(j48) classifier achieved the lowest values for Multiclasses classification: 0.0796 error rate. That indicates a reduced number of errors using meta.Adaboost(j48)for Multiclasses classification. The MLP classifier achieved the highest values for Multiclasses classification: 0.2119 error rate. This means that this classifier achieved the highest number of errors as compared to the other classifiers in the classification process. We can also see that the Random Forest classifier achieved a percentage convergent with the lowest error rate: 0.0835%. The error rate for J48 classifier was closer to the rate for Random Forest classifier: 0.1101%.

The area under the ROC curves for meta.Adaboost (j48), Random Forest, J48 and MLP classifiers are shown in Figure 4.27, Figure 4.28, Figure 4.29 and Figure 4.30respectively for Multiclasses classification.

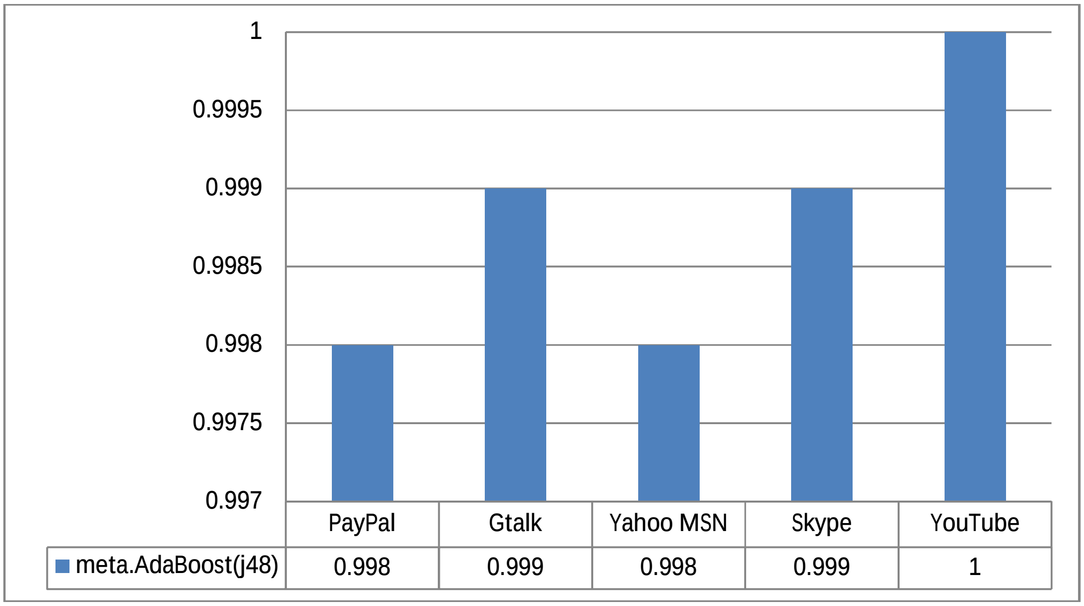


Figure 4.27
Area under ROC for Multiclasses of meta.Adaboost(j48).

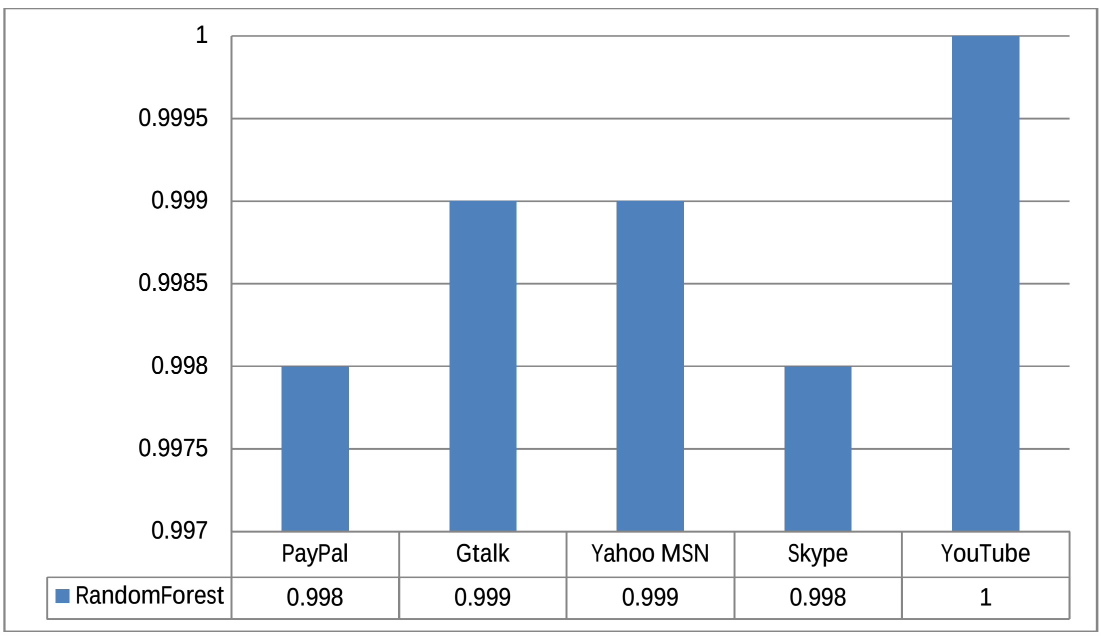


Figure 4.28
Area under ROC for Multiclasses of Random Forest.

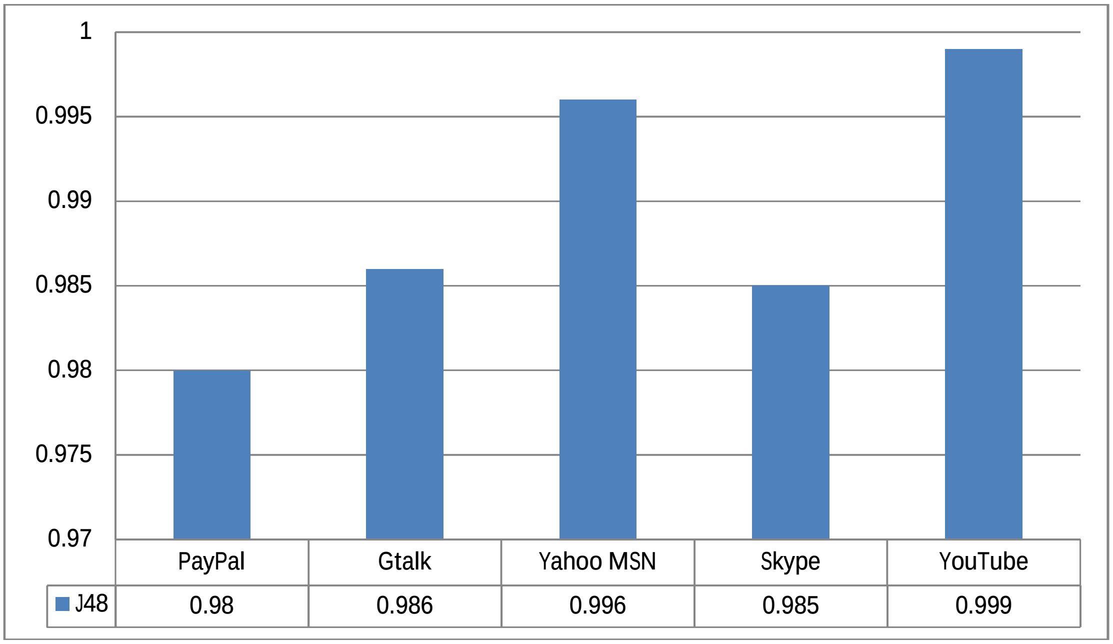


Figure 4.29
Area under ROC forMulticlassesofJ48.

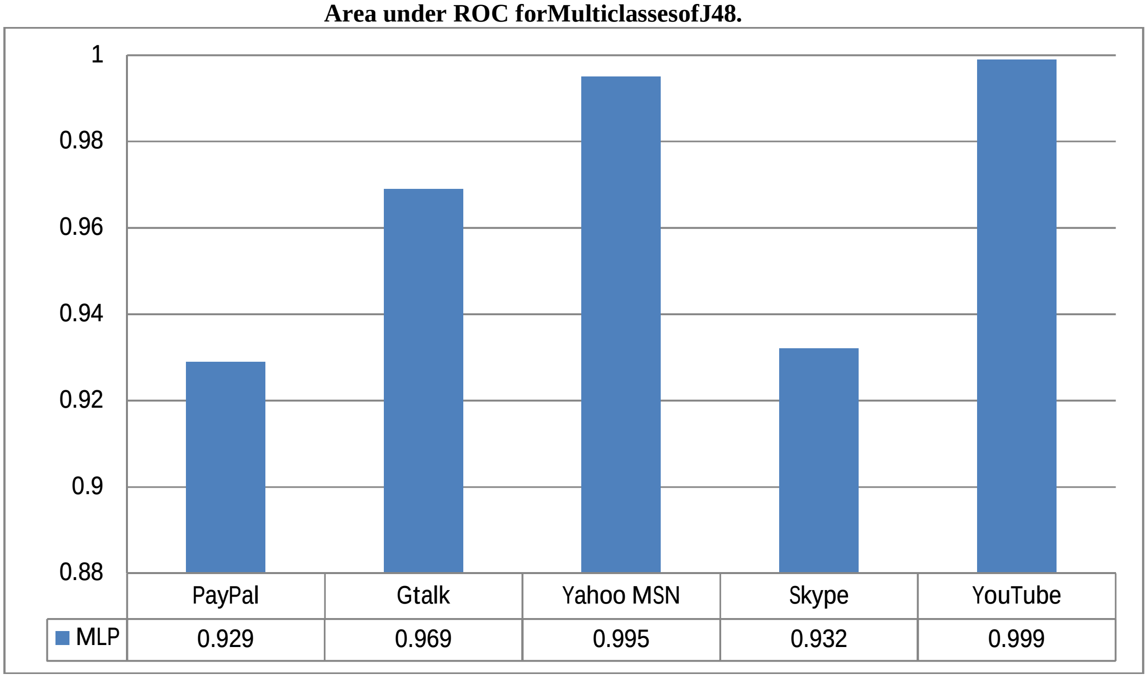


Figure 4.30
Area under ROC for Multiclasses of MLP.

We analyzed the result of the ROC curve in detail as follows. The YouTube application achieved excellent prediction in the ROC curve: 1 ROC curve values, which indicates full values in both meta.Adaboost(j48)and Random Forest. In the other classifiers, J48 and MLP, YouTube achieved the highest values of all the other classes: 0.999. In the meta. Adaboost (j48)classifier, the PayPal and Yahoo Messenger applications achieved the lowest values of the ROC curve, which means a

poor prediction in these cases, and the Gtalk and Skype application values were in the middle range of prediction.

In the Random Forest classifier, the PayPal and Skype applications achieved the lowest values of the ROC curve, which means a poor prediction in these instances, and the Gtalk and Yahoo Messenger application values were in the middle range of prediction. In the J48 classifier, the PayPal application achieved the lowest values of the ROC curve, which means a poor prediction in these instances, and the Gtalk and Skype application values were in the middle range of prediction. The percentage of the ROC curve for the Yahoo Messenger application in the J48 classifier was closer to the highest rate that was achieved by the YouTube application.

Finally, in the MLP classifier, the Yahoo Messenger application was also closer the highest rate that was achieved by the YouTube application. The Gtalk application came after the Yahoo Messenger rate, and Skype and PayPal achieved the lowest values of all the other classes, their results equaling 0.932 and 0.929 respectively, but PayPal represents the worst prediction rate for the ROC curve. In general, we can see from the previous figures that there was no big difference between application values in each classifier.

The time that ML classifiers need to build a training model using the WEKA toolbox is illustrated by Figure 4.31. This shows the time that the selected classifier takes to build a training model for Multiclasses classification.

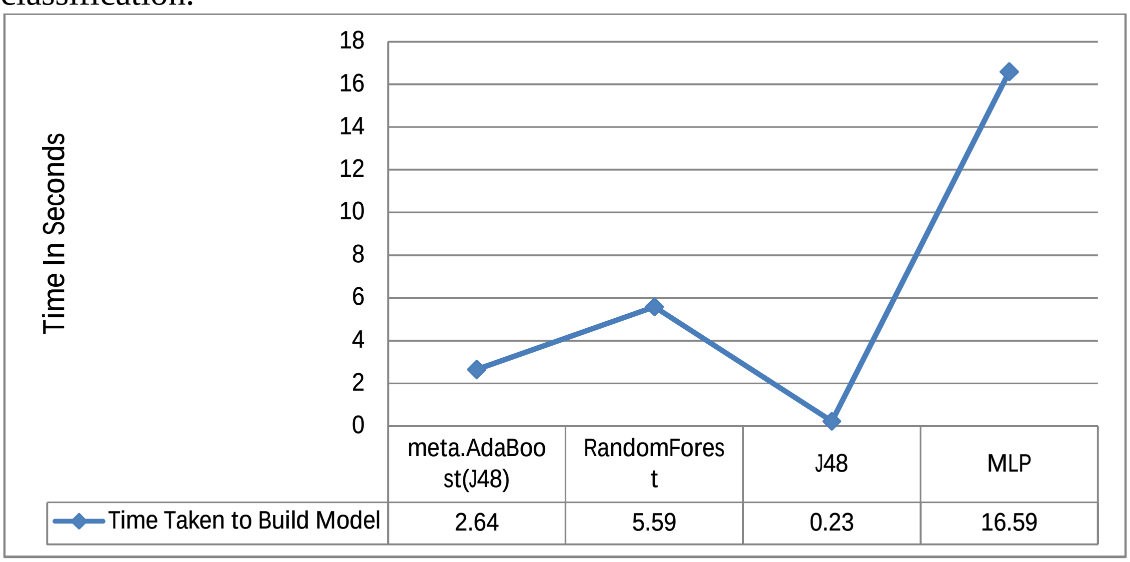


Figure 4.31
Time Taken to Build a Model for Multiclasses.

The J48 classifier took the shortest time to build a model for our generated dataset for Non-VOIP and VOIP classification: 0.23 seconds, which is the fastest time among all the classifiers. On the other hand, MLP

took the longest time to build a model in our dataset for Multiclasses classification: 16.59 seconds, which is the slowest time among all the ML classifiers.

As regards the other classifiers, meta.Adaboost(j48) achieved second place after the J48 classifier with a time for Multiclasses classification of 2.64 seconds. Random Forest achieved third place after meta.Adaboost(j48) and J48's classifiers time to build a model placed it in order before the MLP classifier, with a time of 5.59 seconds.

4.4 Outcomes Summary

In summary, the results above for Non-VOIP and VOIP and Multiclasses classification cases are based mainly on precision and recall rate and many other metrics mentioned earlier. These metrics were used to measure the performance of the four ML algorithms within two classification cases, namely: Non-VOIP and VOIP classification case which contained two classes, each class containing two or more applications according to their classification such that the Non-VOIP class included the PayPal and YouTube applications, and the second class included the Skype, Yahoo Messenger and Gtalk classes. The other case of classification was Multiclasses, which included five classes, each class represent application, namely: PayPal, YouTube, Skype, Yahoo Messenger and Gtalk classes. Each case was tested using a 10-fold cross validation mode from the WEKA toolbox.

The following Figure 4.32 and Figure 4.33 represent a comparison based on the precision rate for each ML classifier shown twice: once for Non-VOIP and VOIP classification case and the other for Multiclasses classification case.

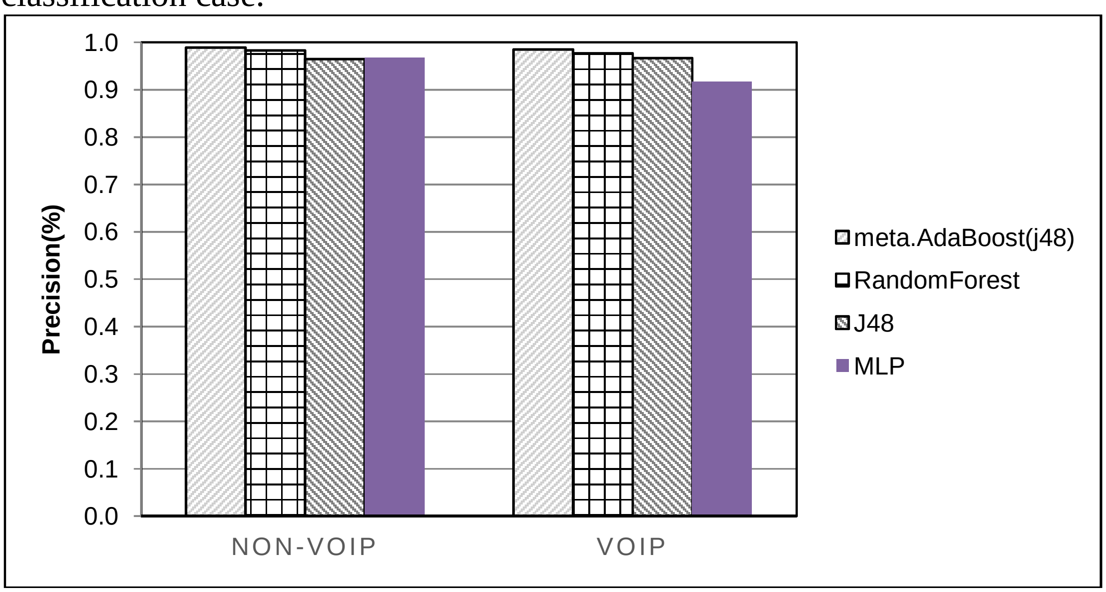


Figure 4.32
Precision Rates for ML Classifiers in Non-VOIP and VOIP Classification.

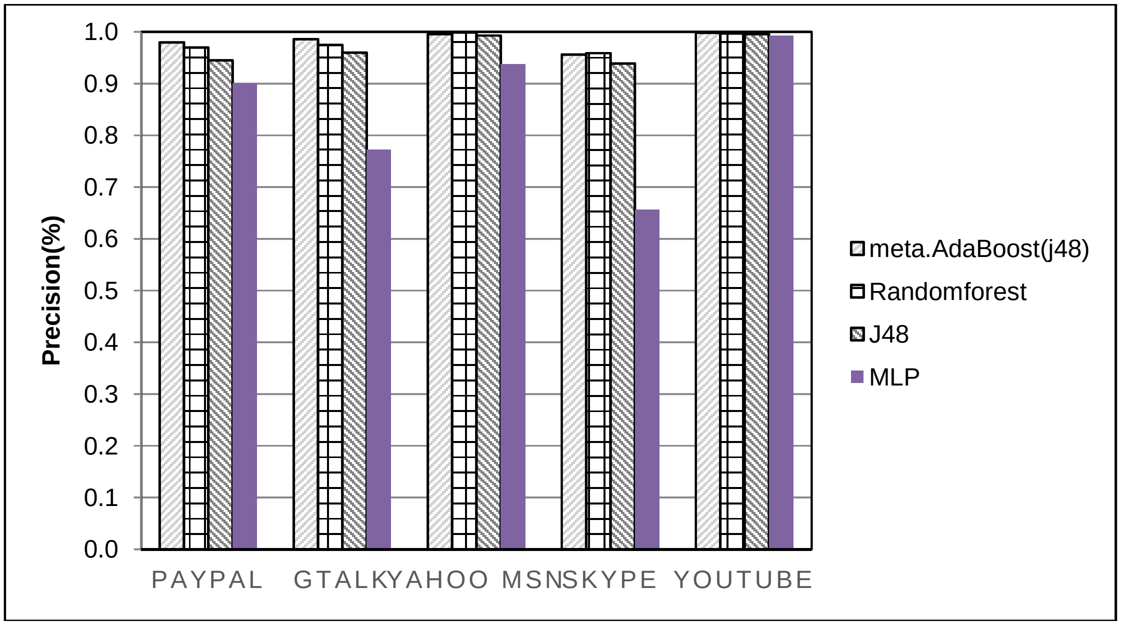


Figure 4.33
Precision Rates for ML Classifiers in Multiclasses Classification.

Figure 4.32 and Figure 4.33 show precision percentages for each ML classifier within Non-VOIP and VOIP and Multiclasses classification cases. The highest precision rate was achieved by the meta.Adaboost(j48) classifier for each class in Non-VOIP and VOIP case. On the other hand, the same classifier achieved the highest precision rate for each class of Multiclasses case except for the Yahoo Messenger and Skype classes, with a small difference for the Random Forest classifier that achieved the highest precision rate.

For Non-VOIP and VOIP classification case the Random Forest classifier achieved second place. For Multiclasses classification case Random Forest classifier got the second place for the PayPal, Gtalk and YouTube classes. The meta. Adaboost (j48) classifier achieved second place in the Yahoo Messenger and Skype classes, and there was no big difference as compared to the previous precision rates. In addition, the J48 classifier for Non-VOIP and VOIP classification achieved fourth place after the MLP classifier in the Non-VOIP class, which means that it achieved the lowest precision rate for Non-VOIP, and third place for the VOIP class. The MLP classifier came after the J48 classifier, which means that it achieved the lowest precision rate for VOIP class. For Multiclasses classification, the J48 classifier achieved third place in all classes. The MLP classifier achieved the lowest precision rate in all classes.

Recall rate is another important metric for measuring the performance of the ML classifiers used. Figure 4.34 and Figure 4.35 show recall rates for each ML classifier within the two classification types, including their classes. Later we shall explain the comparison between them in more detail.

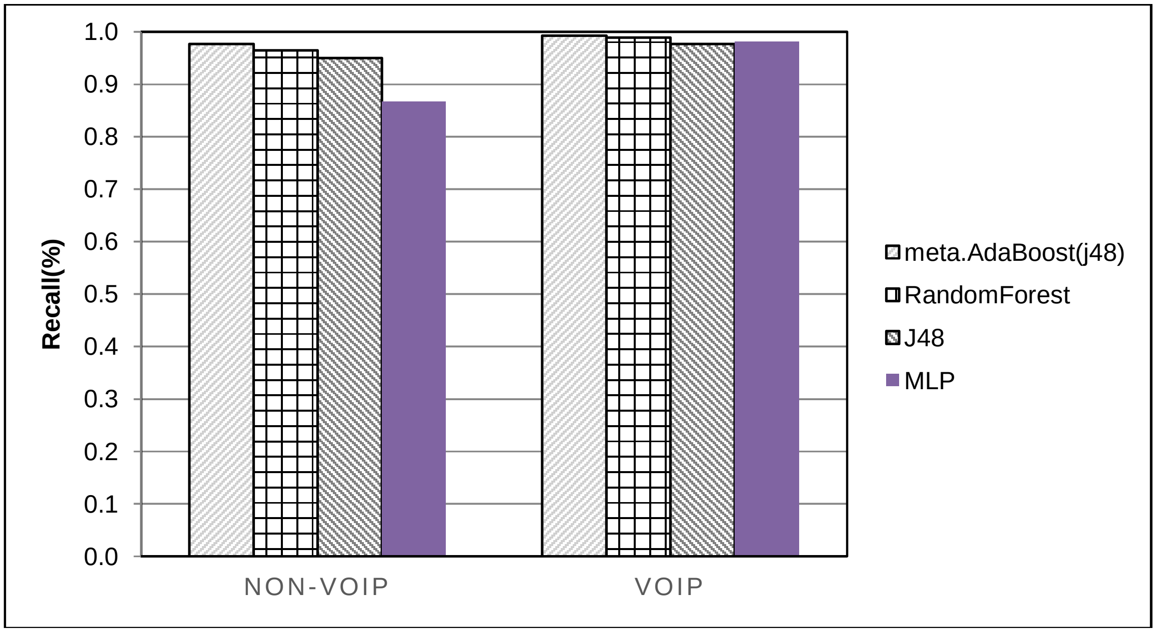


Figure 4.34
Recall Rates of ML Classifiers in Non-VOIP and VOIP Classification.

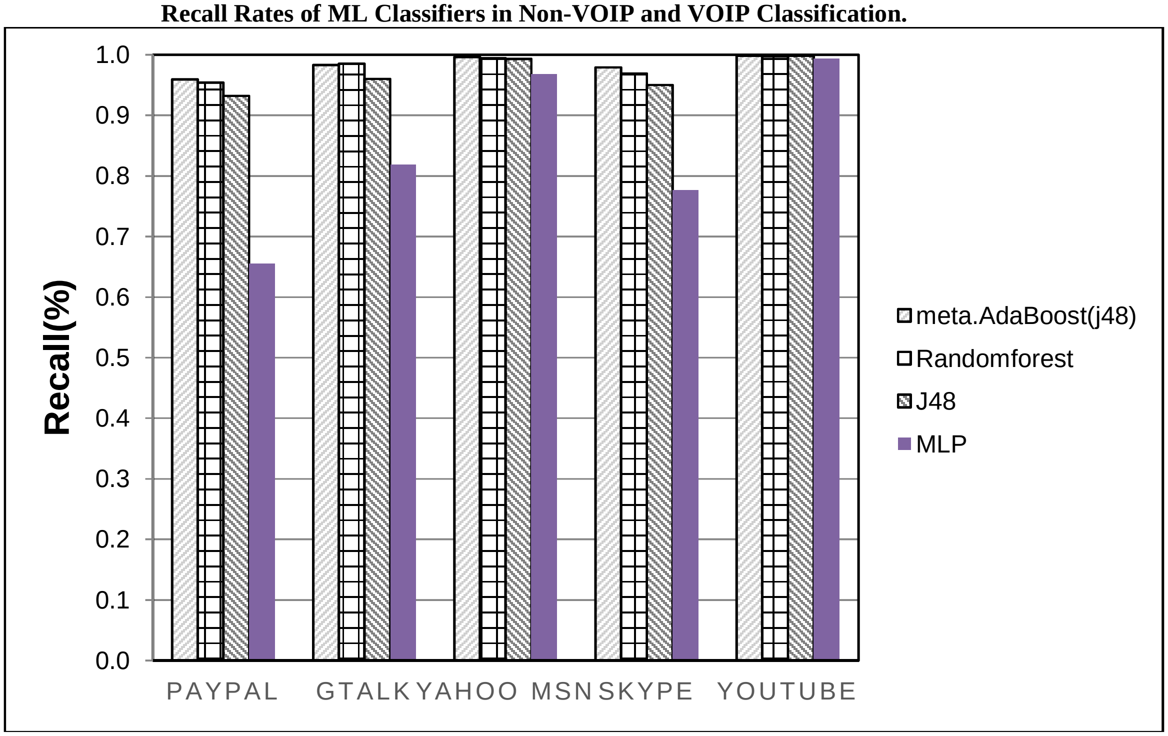


Figure 4.35
Recall Rates of ML Classifiers in Multiclasses Classification.

In the previous two figures the recall percentages of each ML algorithm in the two classification cases, including their classes, were represented. In Non-VOIP and VOIP classification case the meta.Adaboost(j48) classifier achieved the highest recall rate for the correct classification from the instances that were actually positive for their classes. The same classifier also achieved the highest recall rate for Multiclasses except for the Gtalk class with only a small difference. The

YouTube class of the J48 classifier achieved the same rate as for meta.Adaboost(j48): 0.998.

On the other hand, the Random Forest classifier achieved second place for all classes within Non-VOIP and VOIP classification case. In addition, it also gained second place for all classes within Multiclasses classification case except for the Gtalk class which achieved the highest recall rate with only a small difference, as mentioned earlier. The J48 classifier gained third place for the Non-VOIP class but not for the VOIP class, while the MLP classifier gained third place for the VOIP class within the Non-VOIP and VOIP classification. For Multiclasses classification case, J48 classifier achieved third place for all classes except for the YouTube class, as previously mentioned.

For the final classifier, MLP achieved the lowest recall rate for the Non-VOIP class only, and third place for the VOIP class. The J48 classifier achieved the lowest recall rate for the VOIP class within Non-VOIP and VOIP classification case, while the MLP classifier achieved lowest recall rate for all classes within Multiclasses classification case. In conclusion, the YouTube class achieved very convergent rates for all ML classifiers. In addition, we can see that there was no big difference between the values of recall within the two classification types.

The final accuracy rate represents a very important criterion with which to measure the performance of the ML algorithms between each other and between the two classification cases. Thus, we can show the difference between them.

The meta. Adaboost(j48) classifier achieved the highest accuracy of all the classifiers for each case of Non-VOIP and VOIP and Multiclasses classification case. However, in the case of Non-VOIP and VOIP, the accuracy rate was greater than the accuracy rate in the case of Multiclasses for the same classifier. Their accuracy rate was equal to 98.6605% and 98.3007% for Non-VOIP and VOIP case and Multiclasses classification case respectively.

The Random Forest classifier gained second place in the accuracy rate for each classification cases, but the Multiclasses case achieved an accuracy rate greater than the accuracy rate for the Non-VOIP and VOIP case for the same classifier with only a small difference. Their result was equal to 97.9405% and 97.9804% for Non-VOIP and VOIP case and Multiclasses classification case respectively.

The J48 classifier achieved third place for accuracy rate in both classification cases. The accuracy rate for the Non-VOIP and VOIP case was less than the Multiclasses case accuracy rate. Their results showed the small difference between them equal to 96.6413% and 96.6613% for Non-VOIP and VOIP case and Multiclasses classification case respectively. For

the above classifiers there was no big difference between their results in Non-VOIP and VOIP case and Multiclasses classification case.

The MLP classifier achieved the lowest accuracy rate among all the classifiers in each case. It achieved results in Non-VOIP and VOIP classification case much better than Multiclasses classification, showing a great difference between the results, equal to 93.5426% and 84.1663% for Non-VOIP and VOIP case and Multiclasses classification case respectively.

4.5 Conclusions and Future Work

Classification of network traffic is most important nowadays, and it is a sensitive issue due to the widespread availability of interactive and online applications. Most people use interactive applications such as Skype, Gtalk and Yahoo Messenger as VOIP applications. Other Non-VOIP applications that are commonly used include YouTube and PayPal. All of these applications may be gathered from the TL within TLS, SSL, TCP, UDP and HTTPs protocols that provides data integrity and privacy for all the data that crosses a network between two communication nods.

In this thesis, we generated a new real dataset for five different applications captured from real-life network traffic. It is includes five important and different applications. We carried out a classification in two cases: Non-VOIP and VOIP classification case for two classes separated as follows: Non-VOIP (YouTube and PayPal) and VOIP (Skype, Gtalk, and Yahoo Messenger). The second case dealt with classification of the five applications in detail that were called Multiclasses case, containing five classes represented as follows in our dataset: PayPal, YouTube, Gtalk, Yahoo Messenger, and Skype. We collected this dataset from experimental environment within a campus environment based on four important statistical features. These features were chosen from many other features according to their importance for the performance accuracy rate. In addition, we tested our dataset on four different ML algorithms using the WEKA toolbox.

The proposed ML classifiers were meta.Adaboost (j48), Random Forest, J48 and MLP. They tested data models using the new generated dataset. A comparison was made between them and the results showed that the meta.Adaboost (j48) algorithm achieved the highest accuracy result for Non-VOIP and VOIP case and Multiclasses classification case equal to98.6605% and 98.3007% respectively. Random Forest achieved 97.9408% and 97.9808 % accuracy rate, quite close to the highest accuracy rate for both Non-VOIP and VOIP case and Multiclasses classification case respectively.

The J48 decision tree achieved 96.6413% and 96.6613% respectively that were very close to the Random Forest accuracy rate. The

MLP classifier achieved the lowest accuracy rate for both Non-VOIP and VOIP case and Multiclasses classification case respectively, equal to 93.5426% and 84.1663%, which represent a large difference from the results of the classifiers that were mentioned previously. The result of Non-VOIP and VOIP classification case is better than that of Multiclasses classification case, as mentioned previously regarding percentages. Therefore, we conclude here that some classifiers are capable of achieving excellent results, while other classifiers achieve bad results. Here the meta.Adaboost (j48) classifier achieved excellent results, the MLP classifier had bad result and the results for the other classifiers were in the middle range. These results represent both the Non-VOIP and VOIP case and Multiclasses classification case respectively.

In respect of future work, we recommend focusing on other VOIP and non-VOIP applications that are used the most. A new dataset could be generated that separates the classes into only two parts, such as VOIP and Non-VOIP, or to be even more detailed, separate it according to the application numbers, such as in our dataset. In addition, the dataset could be tested using other classifiers according to the requirements of the research.

In addition, the number of features could be increased according to traffic classification needs and to achieve a high level of accuracy for the performance of the ML classifiers. Other statistical features could be used, such as min, mean, max, STD, as well as other features. Another tool could be used to capture data on another platform, such as the Netmate tool that is installed on the Linux platform.

Other researchers could apply methods for protection based on the dataset generated or develop new methods depending on the dataset to protect important and personal data over the network. In addition, data could also be collected from other platforms or simulators, such as NS-2 and OpenNet.

REFERENCES

Abderrahim, H., Chellali, M. R., & Hamou, A. (2015). Forecasting PM10 in Algiers: efficacy of multilayer perceptron networks. **Environmental Science and Pollution Research,** 1-8.

Adami, D., Callegari, C., Giordano, S., Pagano, M., & Pepe, T. (2012). Skype-Hunter: A real-time system for the detection and classification of Skype traffic. **International Journal of Communication Systems**, 25(3), 386-403.

Alshammari, R., & Zincir-Heywood, A. N. (2011). Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?.**Computer networks**, 55(6), 1326- 1350.

Alshammari, R., & Zincir-Heywood, A. N. (2015). Identification of VoIP encrypted traffic using a machine learning approach. **Journal of King Saud University-Computer and Information Sciences**, 27(1), 77-92.

Asrodia, P., & Patel, H. (2012). Analysis of various packet sniffing tools for network monitoring and analysis. **International Journal of Electrical, Electronics and Computer Engineering**, 1(1), 55-58.

Behrouz A.Forouzan . (2004). **Data communication and Networking**. Prentice-Hall, 5th Edition

Bujlow, T., Riaz, T., & Pedersen, J. M. (2012, January). A method for classification of network traffic based on C5. 0 Machine Learning Algorithm. In **Computing, Networking and Communications (ICNC), 2012 International Conference on (pp. 237-241). IEEE.**

Dainotti, A., Pescape, A., & Claffy, K. C. (2012). Issues and future directions in traffic classification. **Network, IEEE**, 26(1), 35-40.

Del Río, P. S., Ramos, J., García-Dorado, J. L., Aracil, J., & Cutanda-Rodríguez, M. (2011, July). On the processing time for detection of Skype traffic. In **Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International** (pp. 1784-1788). IEEE.

Duan, Q., & Al-Shaer, E. (2013). Traffic-aware dynamic firewall policy management: techniques and applications. **Communications Magazine, IEEE**,51(7), 73-79.

Fonseca, H., Cruz, T., Simoes, P., Monteiro, E., Silva, J., Gomes, P., & Centeio, N. (2014, September). A comparison of classification techniques for detection of VoIP traffic. In **Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014 Eighth International Conference on** (pp. 117-122). IEEE.

Ibrahim, H. A. H., Nor, S. M., Mohammed, A., & Mohammed, A. B. (2012). Taxonomy of machine learning algorithms to classify real time interactive applications. **International Journal of Computer Networks and Wireless Communications**, 2(1), 2012.

Jagtap, S., & G., K. (2013). Census Data Mining And Data Analysis Using WEKA. **International Conference in Emerging Trends in Science, Technology and Management**, (pp. 35-40). Singapore.

Kevric, J., & Subasi, A. (2012). **Classification of EEG signals for epileptic seizure prediction using ANN**.

Li, F., Claypool, M., & Kinicki, R. (2015, February). Treatment-based traffic classification for residential wireless networks. In **Computing, Networking and Communications (ICNC), 2015 International Conference** on (pp. 160-165). IEEE.

Mahajan, V. S., & Verma, B. (2012, December). Implementation of network traffic classifier using semi supervised machine learning approach. **In Engineering (NUiCONE), 2012 Nirma University International Conference** on(pp. 1-6). IEEE.

Masud, M. M., Mustafa, U., & Trabelsi, Z. (2014, March). A data driven firewall for faster packet filtering. **In Communications and Networking (ComNet), 2014 International Conference** on (pp. 1-5). IEEE.

Needleman, M. (2000). The internet engineering task force. **Serials Review**,26(1), 69-72.

Qin, T., Wang, L., Liu, Z., & Guan, X. (2015). Robust application identification methods for P2P and VoIP traffic classification in backbone networks. **Knowledge-Based Systems**, 82, 152-162.

Shao, Y., Zhang, L., Chen, X., & Xue, Y. (2014, October). Towards time-varying classification based on traffic pattern. In **Communications and Network Security (CNS), 2014 IEEE Conference** on (pp. 512-513). IEEE.

Sinam, T., Singh, I. T., Lamabam, P., Devi, N. N., & Nandi, S. (2014, February). A technique for classification of VoIP flows in UDP media streams using VoIP signalling traffic. In **Advance Computing Conference (IACC), 2014 IEEE International** (pp. 354-359). IEEE.

Tapaswi, S., & Gupta, A. S. (2013, October). Flow-based P2P network traffic classification using machine learning. In **Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2013 International Conference** on (pp. 402-406). IEEE.

Tiwari, A., & Prakash, A. (2014). **Improving classification of J48 algorithm using bagging, boosting and blending ensemble methods on SONAR dataset using WEKA**.

Van Essen, B., Macaraeg, C., Gokhale, M., & Prenger, R. (2012, April). Accelerating a random forest classifier: Multi-core, GP-GPU, or FPGA?. In **Field-Programmable Custom Computing Machines (FCCM), 2012 IEEE 20th Annual International Symposium on (pp. 232-239). IEEE.**

Wicaksana, A., & Sasongko, A. (2011, July). Fast and reconfigurable packet classification engine in FPGA-based firewall. In **Electrical Engineering and Informatics (ICEEI), 2011 International Conference** on (pp. 1-6). IEEE.

Xue, Y., Wang, D., & Zhang, L. (2013, January). Traffic classification: issues and challenges. **In Computing, Networking and Communications (ICNC), 2013 International Conference on** (pp. 545-549). IEEE.

Zhang, J., Xiang, Y., Wang, Y., Zhou, W., Xiang, Y., & Guan, Y. (2013). Network traffic classification using correlation information. **Parallel and Distributed Systems, IEEE Transactions on**, 24(1), 104-117.

Zhao, S., Yu, X., Chen, Z., Jing, S., Peng, L., & Liu, K. (2012, September). A novel online traffic classification method based on few packets. In **Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on** (pp. 1-4). IEEE.

Zhao, Y., & Zhang, Y. (2008). Comparison of decision tree methods for finding active objects. **Advances in Space Research**, 41(12), 1955-1959.

المعلومات الشخصية

الاسم: نسيبة حمدان أبو سمهدانة

التخصص: ماجستير علم الحاسوب

الكلية: العلوم

السنة: 2015